Ladybug SDK Help

Version SDK v1.18

Revised 1/19/2023

FCC Compliance

This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesirable operation.

Korean EMC Certification

The KCC symbol indicates that this product complies with Korea's Electrical Communication Basic Law regarding EMC testing for electromagnetic interference (EMI) and susceptibility (EMS).

Hardware Warranty

The warranty for the camera is 2 Years. For detailed information on how to repair or replace your camera, please contact our Technical Support team.

WEEE

The symbol indicates that this product may not be treated as household waste. Please ensure this product is properly disposed as inappropriate waste handling of this product may cause potential hazards to the environment and human health. For more detailed information about recycling of this product, please contact us.

Trademarks

Names and marks appearing on the products herein are either registered trademarks or trademarks of FLIR Integrated Imaging Solutions, Inc. and/or its subsidiaries.

Licensing

To view the licenses of open source packages used in this product please see What open source packages does Ladybug use?

Table of Contents

1 Getting Started with the Ladybug® Software Development Kit

The Ladybug SDK is a software package designed specifically for use with Ladybug cameras. A license to use the package is included with all cameras. The package is compatible with Microsoft Windows and Linux Ubuntu operating systems and consists of:

- A software development kit (SDK)
- A variety of example programs and affiliated source code
- A device driver
- The LadybugCapPro application, which allows you to control many camera functions without any additional programming.

These components provide users with all of the following:

- A complete and easy-to-use programming library for image capture, processing, saving, and display
- A standard C/C++/C# interface allows for easy integration and use with Ladybug cameras
- Uses DMA to minimize latency and processor requirements
- Allows for multithreaded programming
- Full Application Programming Interface (API) allows for complete camera control
- Example programs with full source code

Note: The Ladybug SDK example programs and software libraries can only be used with Ladybug cameras that have the Ladybug driver installed. Cameras that are installed with a third-party camera driver cannot be used with the Ladybug SDK.

1.1 System Requirements

1.1.1 Recommended Requirements to Support a Ladybug6 camera

Recommended RAM	8 GB for capture and recording / 16 GB for post processing
Recommended Operating System	Windows 10 64-bit or Ubuntu 20.04 64-bit for capture, recording, and post processing ARM64 for capture only
Recommended CPU	11th Gen Intel® Core™ i7 processor
Recommended Compilers	Microsoft Visual Studio 2015 or newer / g++ 9.3.0 or newer

Note: The Ladybug SDK example programs and software libraries can only be used with Ladybug cameras that have the Ladybug driver installed. Cameras that are installed with a third-party camera driver cannot be used with the Ladybug SDK.

Note: Ladybug applications that also use Intel Integrated Performance Primitives (IPP) must use IPP version 6.x.

1.2 Installing the Ladybug Camera System

To install your Ladybug system and associated camera and hardware you will need to complete the following steps in this order:

- Install the PCI card into an available PCI slot.
- Download and install the Ladybug SDK.
- Connect the camera components together.
- Connect the camera system to your computer.
- Test the installation.

Note: Refer to your *Getting Started Manual* for detailed installation instructions.

1.2.1 Installation Directory Information

This section describes the contents of the various directories that are installed. The default directory path is: **C:\Program Files\Teledyne\Ladybug**

\bin or \bin64

The \bin directory contains the precompiled executables that are distributed with the SDK (e.g. LadybugCapPro) and the Dynamic Link Libraries (DLLs) required for Ladybug-based applications to work.

By default, the Ladybug example programs, make a copy of their executable in the \bin directory.

\doc

The \doc directory contains all documentation related to Ladybug such as the *Getting Started Manual* and release notes.

\driver

The \driver directory contains the Ladybug ladybug.sys camera driver. The Ladybug driver must be installed in order to use any of the Ladybug programs or example software.

\include

The \include directory contains all header (.h) files required for software development using the Ladybug library. Each header file is fully documented and contains the most up-to-date descriptions of the functionality provided by the Ladybug API (in most cases more recent than the API descriptions contained in this manual).

\lib or \lib64

The \lib directory contains all of the standard or import libraries for the SDK.

\src

The \src directory contains all of the example source code directories. Example source code is created using Microsoft Visual C++.

1.3 Camera Firmware

Firmware is programming that is inserted into the programmable read-only memory (programmable ROM) of most cameras. Firmware is created and tested like software. When ready, it can be distributed like other software and installed in the programmable read-only memory by the user.

The latest firmware versions often include significant bug fixes and feature enhancements. To determine the changes made in a specific firmware version, consult the Release Notes.

Firmware is identified by a version number, a build date, and a description.

1.3.1 Determining Firmware Version

To determine the firmware version number of your camera:

- In LadybugCapPro, open the Camera Control dialog and click on the Camera Information tab
- Query the Firmware Version register 1F60h

1.3.2 Upgrading Camera Firmware

Camera firmware can be upgraded or downgraded to later or earlier versions using the UpdatorGUI program that is bundled with the Ladybug SDK.

Before upgrading firmware:

- Install the SDK, available from the Ladybug SDK page.
- Download the firmware file from our Spherical Imaging section of the website.

Warning! Do not disconnect the camera during the firmware update process.

To upgrade the firmware using LadybugCapPro:

1. ***Start Menu-->Teledyne Ladybug SDK-->UpdatorGUI3***
2. Select the camera from the list at the top.
3. Click Open to select the firmware file.
4. Click Update.
5. Click Yes to continue.

Related Knowledge Base Articles

Title
Teledyne FLIR machine vision software and firmware version numbering scheme
Determining the firmware version used by my camera
Should I upgrade my camera firmware or software?

1.4 LadybugRecorder

The LadybugRecorder is an application that allows users to record streams with GPS but no other processing and no stitching in situations where the host computer does not have the resources to support the fully functional LadybugCapPro.

To access the LadybugRecorder:

Start Menu-->Teledyne Ladybug SDK-->LadybugRecorder

1.5 Stream File Repair

The Stream File Repair utility attempts to repair stream files that may have become corrupted.

To access the Stream File Repair utility:

Start Menu-->Teledyne Ladybug SDK-->LadybugStreamRepairUtility

1.6 Camera Care

1.6.1 Camera Care

Warning! Do not open the camera housing. Doing so voids the Hardware Warranty.

Your camera is a precisely manufactured device and should be handled with care. Here are some tips on how to care for the device.

- Avoid electrostatic charging.
- When handling the camera unit, avoid touching the lenses. Fingerprints affect the quality of the image produced by the device.
- To clean the lenses, use a standard camera lens cleaning kit or a clean dry cotton cloth. Do not apply excessive force.
- Avoid excessive shaking, dropping or any kind of mishandling of the device.

Note: To replace the protective glass the camera must be returned to Teledyne for servicing. Contact Support for more details.

2 Using the LadybugCapPro Application

The LadybugCapPro application provides an easy-to-use interface for controlling many functions of your Ladybug camera. LadybugCapPro consists of two primary interfaces: the Main Window and the Camera Control Dialog.

Interface	Functions
Main Window	■ Control image processing settings, including color processing algorithm, falloff correction, blending width, projection, stabilization and vertical tilt adjustment. ■ View a live video stream from the camera and record stream files. ■ Output stream files into other formats. ■ Save individual panoramic images. ■ Record positional data from a GPS device into a stream, and generate Google Maps or Google Earth files.
Camera Control Dialog	■ Control settings such as brightness, gain, shutter, white balance and others. ■ Adjust JPEG Compression. ■ Operate the camera in High Dynamic Range mode. ■ Configure the GPIO for trigger/strobe control. ■ Access camera registers. ■ Control each sensor independently for shutter, gain and auto exposure. ■ Access advanced settings.

To start LadybugCapPro

To run LadybugCapPro from the **Start** menu, select **Teledyne Ladybug SDK> LadybugCapPro**.

The Welcome Dialog opens.

2.1 Welcome Dialog

When LadybugCapPro starts, the **Welcome** dialog opens. You have a choice of starting a camera, or loading a previously recorded stream file.

Start Camera

If you choose to start a camera, the **Select Camera** dialog opens. This dialog allows you to view a list of all the currently connected Ladybug cameras across all buses, and select one to control and view images from. The dialog also lists basic information for each camera, such as the serial number and the current firmware version.

To begin grabbing images, select a camera and click **OK**. The LadybugCapPro Main Window opens in live image-grabbing mode.

To access the Camera Control Dialog prior to grabbing images, select a camera and click **Configure Selected**. After configuring the camera, close the Camera Control dialog and click **OK** to begin grabbing images.

Load Stream File

If you choose to load a stream file, the Windows file explorer opens, allowing you to browse for a .pgr stream file to open. After selecting a stream file, the LadybugCapPro Main Window opens in recorded stream mode.

2.2 Working in the LadybugCapPro Main Window

The Main Window is where you do most of your work in LadybugCapPro. After starting a camera or loading a stream file in the Welcome Dialog, the Main Window opens and displays either a live video stream from the current camera or a previously-recorded video stream.

Note: To magnify the display of toolbar icons for improved accessibility, click Settings -> Options on the menu. In the LadybugCapPro Options dialog, at bottom, click Use large icons. Then click OK.

In Stream File mode, the title bar of the main window contains the file path name, serial number, data format, and frame rate for the loaded stream file.

Functions in LadybugCapPro can be accessed via menus or toolbars.

2.2.1 LadybugCapPro Main Menu

In LadybugCapPro, most tasks represented in the top menu bar can also be performed using the LadybugCapPro toolbars. Using the menu bar, you can accomplish the following additional tasks:

Saving Images

In both Live Camera and Stream File mode, you can save the current image to panoramic JPEG or panoramic bitmap format, or as six individual color-processed images, rectified or non-rectified. For more information, see Saving Images.

Downloading the Configuration File

You can download the file that calibrates the sphere radius for stitching panoramic images. To download this file to your 'My Documents' folder, select **File > Save Configuration File**. By default, images are stitched using a sphere radius of 20 meters. To change the sphere radius, see Adjusting Sphere Size for Stitching.

Downloading the Alpha Mask File

You can download the alpha mask files that dictate pixel opaqueness during the blending stage of the stitching process. To download, select **File > Save Alphamask File**. For more information see Overview of the Ladybug Image Stitching Process.

Getting Help

You can get the following information from the Help menu:

- The SDK Help file.
- LadybugCapPro copyright and version.
- Information about the video card on the system that is being used with LadybugCapPro to render images.

2.2.2 LadybugCapPro Status Bar

The status bar at the bottom of the window displays different information, depending on which mode the application is in.

Live Camera mode

- The first (left-most) status pane displays the status of the connection between LadybugCapPro and the camera unit. A red light here indicates a loss of image. Click on the red light to display event statistics with details.
- The second status pane contains GPS positional information.
- The third status pane shows the display rate, which is the rate at which images are being drawn to screen.
- The third status pane shows the actual rate at which the images are being grabbed from the camera.
- The final (right-most) status pane shows the rate by which image data is being transferred from the camera to the PC over the bus.

For more information, see Capturing Stream Files.

Stream File mode

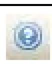
- The first (left-most) status pane contains information about the status of stream conversion.
- The second status pane contains GPS positional information.
- The third status pane shows the rate at which image conversion is processed.
- The fourth status pane shows the index number of the current image being displayed, out of the total number of images in the stream.
- The fifth status pane shows the current values of the left and right keyframes.

- The final (right-most) status pane shows the shutter, gain and gamma settings under which the stream file was recorded.

For more information, see Viewing and Outputting Stream Files.

2.2.3 Main Toolbar

Use the Main Toolbar for connecting to a new camera (or stream) or changing LadybugCapPro application settings.

Icon	Description
	Starts a new camera or loads a .pgr stream file. For more information, see Welcome Dialog.
	Allows you to set the following: ▪ Options for communicating with your GPS receiver. See Working with GPS Data ▪ JPEG Compression Quality--Controls the quality of images that are saved from a stream file in JPEG format. See Saving Images . We recommend the default setting of 85%. The increased file size and processing resources at higher settings may not be worth the minimal increase in quality. ▪ Options for Google Map. See Generating Google Maps and Google Earth data. ▪ Stabilization - Adjusts Parameters for working with image stabilization. See Stabilizing Image Display. ▪ Dynamic Stitch properties used for auto and one shot dynamic stitch. See Adjusting Sphere Size for Stitching. ▪ Use large icons--Check to magnify the display of toolbar icons.
	Copyright information about LadybugCapPro.

2.2.4 Live Camera Toolbar

The Live Camera Toolbar displays only if the camera is in live image-grabbing mode. Use this toolbar for the following functions:

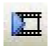
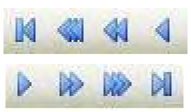
- Start or stop recording a stream.
- Pause the grabbing of images from the camera.
- Access the Camera Control Dialog.
- Change the data format (pixel format) of the images being outputted from the camera.
- Perform a one-shot White Balance auto-adjustment.

- Enable/Disable Independent Auto Exposure.
- Select a Shutter Range from Motion, Indoor, or Low Noise.
- Select an Auto Exposure ROI from Full, Bottom, or Top.

For more information, see Capturing Stream Files.

2.2.5 Stream Navigation Toolbar

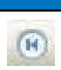
The Stream Navigation toolbar displays only when a previously-recorded stream file is opened. Use this toolbar for navigating within a stream.

Toolbar Control	Description
	Opens a dialog for navigating to a specific frame.
	A series of buttons for navigating through the frames of the video stream. Mouse over each button for an explanation. Alternatively, use the 'Jump to frame' icon or the **Seek** slider at bottom.
	Click to play the stream file. Click again to pause.
	Specifies the first frame from which to begin outputting the stream. Use the buttons at right, or the **Seek** slider at bottom, to navigate to the desired frame. Then click. If not specified, the stream outputs from the beginning frame.
	Specifies the frame on which to stop the output. Use the buttons at right, or the **Seek** slider at bottom, to navigate to the desired frame. Then click. If not specified, the stream output ends at the final frame.

For more information, see Viewing and Outputting Stream Files.

2.2.6 Stream Processing Toolbar

The Stream Processing toolbar display only when a previously-recorded stream file is opened. Use this toolbar for outputting the stream in a different format and resolution.

Toolbar Control	Description
	Sets the left keyframe from which to begin outputting the stream. Use the buttons, or the **Seek** slider at bottom, to navigate to the desired frame. Then click Mark left keyframe. If not specified, the stream outputs from the beginning frame.
	Sets the right keyframe on which to stop the output. Use the buttons, or the **Seek** slider at bottom, to navigate to the desired frame. Then click Mark right keyframe. If not specified, the stream output ends at the final frame.
Output Type	A drop-down list of formats for outputting the stream. For more information about these formats, see Projection Types.

Toolbar Control	Description
Format	The video or image format of the output. **Output Type** (see above) determines which formats are supported. If AVI or H.264 is selected and total output is greater than 2 GB, separate files are created for each sequential 2 GB section of the output.
Output Size	A drop-down list of resolutions for outputting the stream. To specify a custom resolution, select **Custom**.
	**Click to start conversion.** The *Confirm Settings* dialog opens for specifying an **output directory** for the output file. After specifying all applicable settings, click **Convert!** to create the output file(s).
	Click to temporarily stop converting. Click again to resume. If you want to cancel conversion after clicking, click . Any images created before clicking  are saved to the directory you specify, including those created during AVI conversion.
	Click to permanently stop converting. Any images created before clicking are saved to the directory you specify.

For more information, see Viewing and Outputting Stream Files.

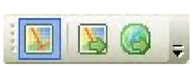
2.2.7 Image Processing Toolbar

The Image Processing Toolbar contains settings that are common to both the live camera and stream file modes. The controls on this toolbar are used to change the way images are processed and rendered. You can use this toolbar to change the color processing algorithm, panoramic viewing angle, panoramic mapping type, falloff correction, blending width, stabilization, sphere size, and color correction. Additionally, you can view a histogram of RGB values represented in the current image.

Control	Description
Color Processing 	Specifies the algorithm that LadybugCapPro uses to convert raw Bayer-tiled image data to 24-bit RGB images. Lower-quality algorithms can increase the LadybugCapPro display rate, and higher-quality algorithms can decrease the display rate. Two additional algorithms are: ▪ **High Quality Linear on GPU**: Same output as High Quality Linear, but better performance on graphics cards with NVidia CUDA support. ▪ **Directional Filter**: Highest quality output, but significantly better performance than Rigorous.
Falloff Correction 	Enables or disables falloff correction, which adjusts the intensity of light in images to compensate for a vignetting effect. This control is off by default. To enable, check **Enable Falloff Correction**. Then, specify an attenuation value either by using the slider or entering a value in the textbox. The attenuation value regulates the degree of adjustment you want to apply. Then click **OK**.

Control	Description
Blending Width	Allows you to adjust the pixel width along the sides of each of the six images within which blending takes place prior to stitching. Blending is the process of adjusting pixel values in each image that overlap with the fields of adjacent images to minimize the effect of pronounced borders. The default width of 100 pixels is suitable for the 20-meter sphere radius to which Ladybug cameras are pre-calibrated. To change the sphere radius calibration, see below.
Image Type	Changes the way images are rendered. See Projection Types. **Note:** These controls affect video display only. To specify how images are rendered when outputting to video, specify an Output Type using the Stream Navigation Toolbar.
Rotation Angle	Specifies the orientation of the camera unit's six cameras to the projection. The default orientation is camera 0 projects to the front of the sphere and camera 5 to the upward pole (or top) of the sphere.
Mapping Type	Specifies the mapping projection that dictates how the six individual pictures from each camera are stitched into a panoramic display—either Radial or Cylindrical. See Projection Types.
Image Stabilization	Adjusts image display to compensate for the effect of unwanted movement across frames when the camera records on an unstable surface. See Stabilizing Image Display.
Sphere Size	Allows you to change the sphere radius, in meters, to which images are calibrated for stitching panoramas. See Adjusting Sphere Size for Stitching.
Image Adjustment	Opens a dialog for performing color correction, sharpening, texture intensity adjustment and tone mapping. See Adjusting 8-bit and JPEG12Processed Images and Adjusting Unprocessed Images.
Anti-Aliasing	Minimizes sampling errors, especially in low-resolution images. From the **Settings** menu, select **Enable Anti-Aliasing**.
Histogram	Displays a histogram of the values represented in the pixels of the current image. **Max Percent** allows you adjust the graphical display to view a subset of percentage representation. For example, to view only the first 5% of the representation of values in the graph, enter '5' in the Max Percent field. **All Camera** specifies that the values are compiled from all six cameras on the Ladybug system. To see values from only one camera at a time, select a camera. (For camera orientation, see **Rotation Angle** above.)

2.2.8 GPS Toolbar



The GPS Toolbar is used for starting or stopping a GPS device, as well as generating Google Map and Google Earth data when a stream file is loaded.

Icon	Description
	Instructs LadybugCapPro to begin receiving positional data from the GPS unit. When used in conjunction with Capturing Stream Files, GPS data is saved with the stream file. For more information, see Stream File Format. This control is not available in recorded stream mode. Click again to stop GPS recording.
	Creates a Google Map file from the GPS data that was previously recorded with the stream file, and allows you the option to load it. An internet connection is required to view the file. Google Maps are saved as .html files in the bin folder of the PGR Ladybug installation directory. This control is not available in live image-grabbing mode.
	Creates a Google Earth file from the GPS data recorded with the stream file, and allows you the option to load it. The Google Earth application and an internet connection are required to view the file. Google Earth files are stored as .kml files in the bin folder of the PGR Ladybug installation directory. This control is not available in image capture mode.

You can also export GPS NEMA data from a loaded stream file using the GPS menu.

For more information, see Working with GPS Data.

2.3 Using the Camera Control Dialog

The Camera Control dialog allows you to control most Ladybug camera functions.

> **Note:** To include this dialog within a custom software application, link to pgrflycapturegui.lib and create a new `CameraGUIContext` within your application. Refer to the LadybugCap demo source code for an example of how to do this.

The following settings can be viewed or set using this dialog:

- Camera Settings - For controlling settings such as Brightness, Exposure, Shutter, Gain and others.
- Camera Information - Provides information about the camera hardware and firmware.
- Camera Registers - Provides direct access to camera registers.
- Trigger / Strobe - For configuring the general purpose input/output (GPIO) capabilities of the camera.
- Advanced Camera Settings - For controlling memory channels, embedded image information and auto-exposure range.
- High Dynamic Range - Enables high dynamic range exposure.
- Data Flash - Provides access to the camera's flash memory.
- System Information - Provides information about the host system to which the camera is connected.
- Bus Topology - Displays the network topology.
- Help / Support - Information about downloading software and firmware updates, accessing the knowledge base, and opening a support ticket.

- Ladybug Settings - For controlling JPEG compression and independent sensor control of exposure settings and auto-exposure statistics.

> **Note:** Some camera controls and formats may be greyed out. If a camera control is greyed out, this means that the camera does not support the function.

2.3.1 Camera Settings

The Camera Settings dialog allows the user to control settings such as Brightness, Exposure, Shutter, Gain and others.

Access to parameters may be limited by which data format is in use. Some images have more control during the image capture phase while some images have more control during post processing.

To open the Camera Settings dialog:

From the Settings menu, select Camera Control and click the Camera Settings tab.

Additional information can also be found in your camera's *Technical Reference Manual*.

2.3.2 Camera Registers

This dialog provides direct access to camera registers, and is therefore recommended for advanced users only. The camera register space conforms to IIDC specifications (see https://www.ieee.org/).

A complete list of CSRs can be found in the *FLIR Machine Vision Camera Register Reference*.

2.3.3 Trigger/Strobe

The GPIO/Trigger dialog provides control over the general purpose input/output (GPIO) capabilities of the camera, including the ability to configure:

- Specific pins for input and output.
- External trigger mode.
- External trigger delay (or shutter delay when not in trigger mode).
- Strobe pulse polarity, duration and delay.

> **Note:** Special output modes such as strobe signal pattern and PWM must be configured using the camera register.

Control	Description
Enable/Disable trigger	When checked, allows the camera to respond to external triggers or internal software triggers.
Mode	Specifies the mode for how the camera responds to an external trigger. Not all modes are supported by all camera models.
Parameter	Certain trigger modes require a parameter to define the triggering cycle.
Trigger Source	Specifies which GPIO pin receives input from an external trigger device.
Trigger Polarity	Specifies a low or high signal polarity.
Trigger Delay	When checked, you can use the slider to specify the time delay, in seconds, from when an external trigger event occurs to the start of integration (when the shutter opens). When **Trigger On/Off** is unchecked, this value represents the shutter delay.
Fire Software Trigger	When clicked, causes a one-time internal (software-based) trigger to fire. **Enable/Disable Trigger** must be checked for the camera to respond.
Pin Direction Control	Specifies whether the pin is configured for input or output. The **Source** pin cannot be configured as an output when **Trigger On/Off** is checked.
Strobe Control Delay (GPIO 0...*n*)	■ Enables a GPIO pin for strobe output. ■ Allows configuration of polarity and the period to delay assertion of the output strobe signal after start of exposure. Delay can be specified in ticks of your camera's clock and must be within the range of 0 to 4095. ■ Specifies the duration of the strobe output signal. If a value of 0 is entered, the duration is the same as the length of exposure. Duration can be specified in ticks of your camera's clock and must be within the range of 0 to 4095.

2.3.4 Advanced Camera Settings

The Advanced Features Dialog allows the control of advanced camera features including:

- User Sets (Memory Channels)
- Ladybug Image Information
- Auto Range Control—Allows you to specify a range for exposure, shutter and gain that is narrower than the full range for the camera, when operating in auto-exposure mode. Use the Camera Settings dialog to set auto-exposure.

2.3.4.1 User Sets (Memory Channels)

The camera can save and restore settings and imaging parameters via on-board user configuration sets, also known as memory channels. This is useful for saving default power-up settings, such as gain, shutter, video format and frame rate, and others that are different from the factory defaults.

User Set 0 (or Memory channel 0) stores the factory default settings that can always be restored. Two additional user sets are provided for custom default settings. The camera initializes itself at power-up, or when explicitly reinitialized, using the contents of the last saved user set. Attempting to save user settings to the (read-only) factory default user set causes the camera to switch back to using the factory defaults during initialization.

The following camera settings are saved in user sets.

- Acquisition Frame Rate and Current Frame Rate
- Image Data Format, Position, and Size
- Image mirror, if applicable
- Current Video Mode and Current Video Format
- Camera power
- Frame information
- Trigger Mode and Trigger Delay
- Imaging Parameters such as: Brightness, Auto Exposure, Shutter, Gain, White Balance, Sharpness, Hue, Saturation, and Gamma
- Input/output controls such as: GPIO pin modes, GPIO strobe modes
- Color Coding ID/Pixel Coding

To access user sets:

- **During Capture**—From the Settings menu, select Camera Control and click the Advanced Camera Settings tab.
- CSRs—Memory Channel Registers

2.3.5 Ladybug Settings

The Ladybug Settings dialog allows you to adjust JPEG compression and control exposure for each sensor independently.

Compression Control

JPEG Quality - Controls the JPEG compression rate of the compressor unit. Increasing the compression rate increases JPEG image quality and, as a result, the amount of image data that is produced and collects in the image buffer.

Select **Auto** to set the compression rate to the maximum allowed by the image buffer. Auto JPEG Quality means the compression rate continually adjusts so that it never exceeds the amount of data allowed by the image buffer. Manual JPEG Quality provides consistent compression however, the size of compressed image data may exceed the image buffer size, resulting in buffer size errors.

A **JPEG Quality** value between 80% and 95% is recommended, depending on your application's requirements. The visual improvement at higher than 95% is negligible and usually not worth the increased amount of data that is generated.

See JPEG Compression and JPEG Buffer Usage.

Auto buffer usage - When **JPEG Quality - Auto** is selected, you can use this slider to specify the percentage of the image buffer that is used for JPEG-compressed image data. Specifying a value less than the maximum allows for room in the image buffer to accommodate extra image data, depending on scene variations from frame to frame. Increasing this value may result in an increase in the **JPEG Quality** setting. When **JPEG Quality - Auto** is not selected, the percentage of the image buffer that is used cannot be controlled.

A **Buffer Usage** setting between 80% and 95% is recommended.

Independent Sensor Control

This interface provides customized control of exposure for each of the six sensors independently for greater dynamic range. Independent Sensor Control is activated by any one of the following ways:

- Selecting the Shutter or Gain **On/Off** control (the On/Off control for each sensor controls all sensors).
- Deselecting the Shutter or Gain **On/Off** control on the Camera Settings pane.
- Clicking the icon on the Live Camera Toolbar.

When shutter or gain is selected in the Independent Sensor Control interface, the following options are available:

- When either shutter or gain is selected, auto exposure can be controlled manually or automatically for each sensor; OR
- When gain is selected, gain can be controlled manually or automatically for each sensor. When shutter is selected, shutter can be controlled manually or automatically for each sensor.

For best results, apply texture intensity adjustment and tone mapping during image processing. For more information, see Adjusting Images.

Sensors Used for Auto Exposure Statistics

When operating in auto exposure mode, you can control which camera sensors are used for calculating the settings of the auto exposure algorithm. For example, if you want all the sensors on the side of the camera to be used in this calculation, but not the top sensor, check boxes 0 through 4, and leave box 5 blank. Leaving all sensors unchecked is equivalent to checking all.

To set exposure in auto mode, use the Camera Settings dialog.

> **Note:** Camera 0 is etched onto the camera housing. Camera 5 is the top sensor.

3 Image Capture Control

Certain parameters are set at the time of image capture.

Further image processing can be done on PC using Post Processing Control .

3.1 Capturing Stream Files

Stream files are saved in the bin folder of the Ladybug SDK installation path.

> **Note:** When capturing stream files, there must be at least 2 GB of free space on the hard drive before writing to disk.

Using LadybugCapPro:

You can record stream files when you start a camera in Live Camera mode, using the controls in the Live Camera Toolbar.

1. From the Settings menu select Camera Control, or click the button.
   - Use the Camera Control dialog to select your trigger mode, and other imaging parameters such as brightness, shutter, gain, auto exposure range, and others.
2. Use the Live Camera Toolbar to start, stop, and pause recording.

Using Ladybug API:

Example writing to disk using the Ladybug API:

1. Create a stream context (`LadybugStreamContext`) by calling `ladybugCreateStreamContext()`.
2. Initialize the stream context for writing by calling `ladybugInitializeStreamForWriting()`.
3. To write an image to disk, simply grab an image, and pass it to `ladybugWriteImageToStream()`.
4. When all the writing is complete, call `ladybugStopStream()` to stop writing to disk.
5. Destroy the context by calling `ladybugDestroyStreamContext()` when suitable (such as program termination).

> **Note:** When used in conjunction with a GPS device, you can record images to stream files when the GPS location changes after a

specified distance. This feature is available using the Ladybug API. For more information, see the ladybugSimpleRecording example.

3.2 Data Formats and Frame Rates

Changing the size of the image or the pixel encoding format requires an undetermined length of frame times, including the stop/start procedure, tearing down/reallocating image buffers, write times to the camera, etc.

Panoramic Image (Full Height Acquisition Mode)	Ladybug6	Ladybug5+
Resolution (MP)	72	30
Dimensions (WxH)	12,288 x 6,144	8,192 x 4,096
Frame Rate (FPS)	15	30
Frame Rate in Standard External Trigger Mode 0 (FPS)	13.9	25.6
Frame Rate in Overlapped Exposure Readout Trigger Mode 14 (FPS)	14.9	30
Panoramic Image (Half Height Acquisition Mode)	**Ladybug6**	**Ladybug5+**
Resolution (MP)	36	15
Dimensions (WxH)	6,144 x 6,144	4,096 x 4,096
Frame Rate (FPS)	29.9	60
Frame Rate in Standard External Trigger Mode 0 (FPS)	25	45
Frame Rate in Overlapped Exposure Readout Trigger Mode 14 (FPS)	28.7	60

Ladybug3 Supported image formats

Data Format	Frame Rate		Image Size	
	Full 1616 x 1232	Half 1616 x 616	Full 1616 x 1232	Half 1616 x 616
Raw8	6.5	13	12 MB	6 MB
JPEG8 Compressed	16	32	Variable	Variable

Ladybug sensors are arranged in "portrait" orientation to increase the vertical field of view. As a result, height measurements appear as width in the Ladybug SDK, and width measurements appear as height.

The image size accounts for six separate images captured by each of the camera's six sensors prior to blending and stitching. Image size for JPEG compressed images is dependent on variables such as image composition and compression rate. For more information, see JPEG Compression and JPEG Buffer Usage.

Related Knowledge Base Articles

Title
Overview of multithreading optimizations in Ladybug library
Ladybug JPEG Image Quality and Buffer Size Settings

Determining Image Size

For Ladybug6, the maximum size of a single camera image after image conversion is 4096 x 2992.

For Ladybug5+, the maximum size of a single camera image after image conversion is 2464 x 2048.

For Ladybug5, the maximum size of a single camera image after image conversion is 2448 x 2048.

For Ladybug3, the maximum size of a single camera image after image conversion is 1616 x 1232.

If your software allocates its own memory for image conversion and texture updating, the amount of memory to be allocated is:

Image Size in MB = (Number of cameras x W x H x BPP) / 1000000

Bytes per pixel (BPP) is 1 for 8-bit modes and 2 for 12-bit modes.

For example, the memory size allocation required for a JPEG8 image after conversion is:

Ladybug6

Image Size = (Number of cameras x W x H x BPP) / 1000000
Image Size = (6 x 4096 x 2992x 1) / 1000000
Image Size = 73531392 / 1000000
Image Size = 73 MB

Ladybug5+

Image Size = (Number of cameras x W x H x BPP) / 1000000
Image Size = (6 x 2464 x 2048 x 1) / 1000000
Image Size = 30277632 / 1000000
Image Size = 30.3 MB

Ladybug3

Image Size = (Number of cameras x W x H x BPP) / 1000000
Image Size = (6 x 2448 x 2048 x 1) / 1000000
Image Size = 30081024 / 1000000
Image Size = 30 MB

Determining Bandwidth

To calculate your bandwidth requirements, use your required resolution, frame rate, and data format as follows:

Bandwidth in MB/s = (Number of Cameras x W x H x FPS x BPP) / 1000000

For example, a JPEG8 full size image at maximum frame rate would use the following bandwidth:

Ladybug6

Bandwidth = (Number of Cameras x W x H x FPS x BPP) / 1000000
Bandwidth = (6 x 4096 x 2992 x 8 x 1) / 1000000
Bandwidth = 588251136 / 1000000
Bandwidth = 588 MB/s

Ladybug5+

Bandwidth = (Number of Cameras x W x H x FPS x BPP) / 1000000
Bandwidth = (6 x 2464 x 2048 x 8 x 1) / 1000000
Bandwidth = 242221056 / 1000000
Bandwidth = 242 MB/s

Ladybug3

Bandwidth = (Number of Cameras x W x H x FPS x BPP) / 1000000

Bandwidth = (6 x 2448 x 2048 x 8 x 1) / 1000000
Bandwidth = 240648192 / 1000000
Bandwidth = 241 MB/s

Determining Frame Rate

The theoretical frame rate (FPS) that can be achieved can be calculated as follows:

Frame Rate in FPS = (Bandwidth / (W x H x BPP)) / Number of Cameras

For example, assuming a JPEG8 full size image, using 240 MB/s bandwidth, the calculation would be as follows:

Ladybug6

Frame Rate = (Bandwidth / W x H x BPP)) / Number of Cameras
Frame Rate = (240000000 / (4096 x 2992 x 1)) / 6
Frame Rate = 3.2 FPS

Ladybug5+

Frame Rate = (Bandwidth / W x H x BPP)) / Number of Cameras
Frame Rate = (240000000 / (2464 x 2048 x 1)) / 6
Frame Rate = 7.93 FPS

Ladybug3

Frame Rate = (Bandwidth / W x H x BPP)) / Number of Cameras
Frame Rate = (240000000 / (2448 x 2048 x 1)) / 6
Frame Rate = 7.98 FPS

3.2.1 Data Formats

Data formats are an encoding scheme by which color or monochrome images are produced from raw image data. Most data formats are numbered to represent the number of bits per pixel.

The Ladybug's Analog-to-Digital Converter, which digitizes the images, is configured to a fixed bit output (12-bit). If the format selected has fewer bits per pixel than the ADC output, the least significant bits are dropped. If the format selected has greater bits per pixel than the ADC output, the least significant bits are padded and can be discarded by the user.

Data Format	Bits per Pixel
Raw 8, JPEG 8	8
Raw 12, JPEG 12	12
Raw 16	16

3.2.1.1 Raw

Raw is a format where image data is Bayer RAW untouched by any on board processing. Selecting a Raw format bypasses the FPGA/color core, which may disable image processing, but allows for faster frame rates. In a GenICam application, this is achieved using Image Format Control and disabling On Board Color Processing.

3.2.1.2 JPEG

JPEG is a format which supports 16.7 million colors and follows a standard for compression by disposing of redundant pixels. The degree of compression can be adjusted allowing for a balance between image size and image quality.

3.2.2 JPEG Compression and JPEG Buffer Usage

When the camera operates in a JPEG-compressed imaging mode, the compressor unit processes image data based on a compression rate. This can be automatically calculated (recommended mode) or user specified. Specifying a higher JPEG quality value produces higher-quality images, but these images are also larger in size. The maximum image size is limited to what the maximum USB3 bandwidth can handle. If the bandwidth is exceeded then frames may be dropped.

Compression and image quality are highly dependent on the scene being captured and can be affected by levels of light, complexity of scene, and data format. Using the JPEG compression in auto mode is recommended to allow the compressor unit to determine the best compression rate to fit the bandwidth and avoid dropped frames.

JPEG compression adjusts to the following parameters:

- The maximum allowed by the size of the image buffer on the PC (controlled by the camera driver).
- Auto-buffer usage. When JPEG compression is in auto mode this setting is the percentage of the image buffer that is used for image data. Specifying a value less than the maximum 100% allows for room in the buffer to accommodate extra images, depending on scene variations from frame to frame. A setting of no more than 95% is recommended. The visual improvement in compression quality that results from a setting higher than 95% is negligible compared to the increased amount of data generated.
- Maximum JPEG quality. When JPEG compression is in manual mode this setting is the maximum percentage for the JPEG compression quality. The recommended value for Ladybug6 is dependent on the data format:

Data Format	%
JPEG8	90
JPEG12	35
JPEG12 Processed	12

To adjust the compression control:

1. From the Settings menu, select Camera Control, or click the button.
2. Click the Ladybug Settings tab.
3. Under Compression Control:

- Select Auto for JPEG Quality to specify auto compression. **This is the recommended setting.**
- Then select an Auto Buffer Usage percentage from the sliding scale. A setting of no more than 95% is recommended.

  Or,
- Deselect Auto JPEG Quality and select a fixed JPEG Quality percentage from the sliding scale.

Using Registers:

- Use the JPEG_CTRL: 1E80h register to turn auto compression on or off and to set a manual JPEG quality percentage.

- Use the JPEG_BUFFER_USAGE: 1E84h register to set the image buffer percentage used if auto JPEG quality is enabled. A setting of no more than 95% is recommended.
- Use the JPEG_MAX_QUALITY: 1E8Ch register to change the maximum JPEG quality percentage used when auto compression is disabled. The default is 80.

Related Knowledge Base Articles

Title
Ladybug JPEG image quality and buffer size settings

3.3 Asynchronous Trigger Settings

Using LadybugCapPro:

You can control the trigger in the Camera Settings:

1. From the Settings menu, select Camera Control, or click the button.
2. Click the Trigger / Strobe tab.
3. Under Trigger Control:
   - Select Enable/Disable trigger.
   - Select a trigger mode from the drop-down list.
   - Enter a parameter.
4. Under Trigger Delay:
   - Select Enable/Disable delay.
   - Use the sliding scale or enter a value for the trigger delay. The trigger delay controls the delay between the trigger event and the start of integration (shutter open).
5. To use a software trigger, click the Fire Software Trigger button.

Using Registers:

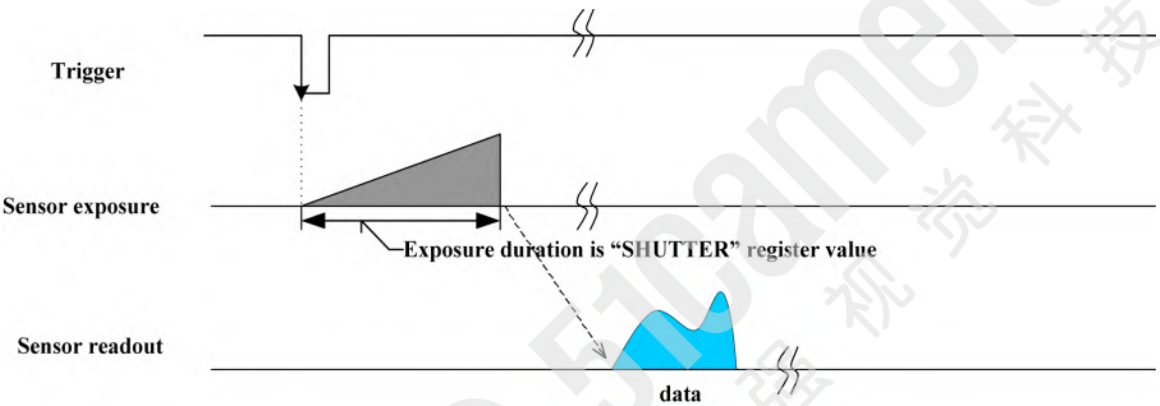
- Use the TRIGGER_MODE: 830h register to enable and select a trigger mode and set parameters.
- Use the TRIGGER_DELAY: 834h register to enable and set a trigger delay.
- Use the SOFTWARE_TRIGGER: 62Ch register to generate a software trigger.

3.4 Standard External Trigger (Mode 0)

Trigger Mode 0 is best described as the standard external trigger mode. When the camera is put into Trigger Mode 0, the camera starts integration of the incoming light from external trigger input falling/rising edge. The Shutter value describes integration time. No parameter is required. The camera can be triggered in this mode by using the GPIO pins as external trigger or by using a software trigger.

It is not possible to trigger the camera at full frame rate using Trigger Mode 0; however, Overlapped Exposure Readout Trigger (Mode 14) allows triggering at nearly full frame rate.

	Full Height	Half Height
Ladybug6 Frame Rate	13.9	25
Ladybug5+ Frame Rate	25.6	45

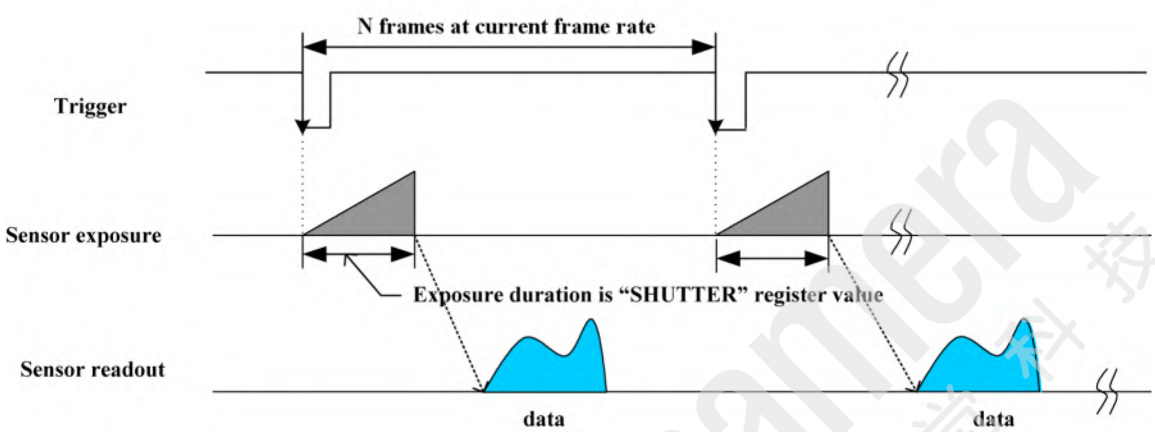


Trigger Mode 0 ("Standard External Trigger Mode")

Registers—TRIGGER_MODE: 830h		
Presence	[0]	1
ON	[6]	1
Polarity	[7]	Low/High
Source	[8-10]	GPIO Pin
Value	[11]	Low/High
Mode	[12-15]	Trigger_Mode_0
Parameter	[20-31]	None

3.5 Skip Frames Trigger (Mode 3)

Skip Frames mode allows the user to put the camera into a mode where the camera only transmits one out of *N* specified images. This is an internal trigger mode that requires no external interaction. Where *N* is the parameter set in the Trigger Mode, the camera will issue a trigger internally at a cycle time that is *N* times greater than the current frame rate. As with Trigger Mode 0, the Shutter value describes integration time.



Trigger Mode 3 ("Skip Frames Mode")

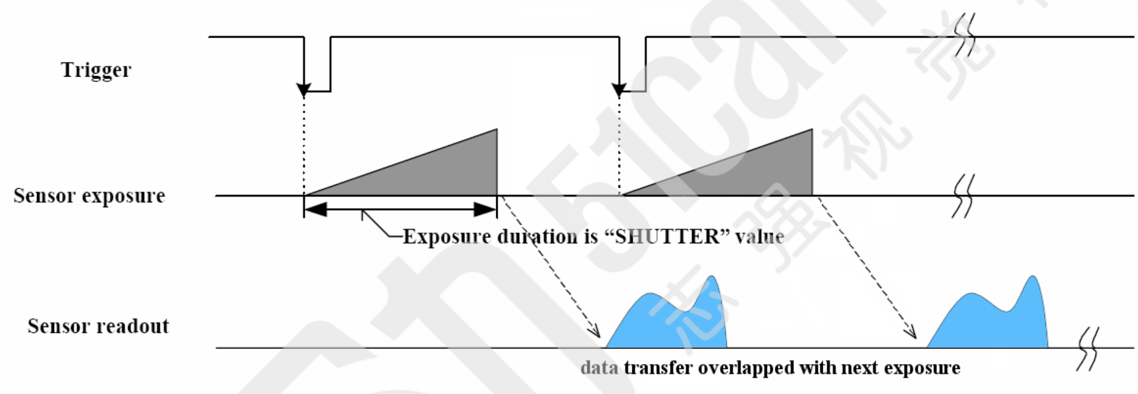
Registers—TRIGGER_MODE: 830h		
Presence	[0]	1
ON	[6]	1
Polarity	[7]	Low/High
Source	[8-10]	GPIO Pin
Value	[11]	Low/High
Mode	[12-15]	Trigger_Mode_3
Parameter	[20-31]	*N* 1 out of N images is transmitted. Cycle time N times greater than current frame rate

3.6 Overlapped Exposure Readout Trigger (Mode 14)

Overlapped Exposure Readout mode is a vendor-unique trigger mode that is very similar to Standard External mode, but allows for triggering at faster frame rates. This mode works well for users who want to drive exposure start with an external event. However, users who need a precise exposure start should use Standard External Trigger (Mode 0).

	Full Height	Half Height
Ladybug6 Frame Rate	14.9	28.7
Ladybug5+ Frame Rate	30	60

In the figure below, the trigger may be overlapped with the readout of the image, similar to continuous shot (free-running) mode. If the trigger arrives after readout is complete, it starts as quickly as the imaging area can be cleared. If the trigger arrives before the end of shutter integration (that is, before the trigger is *armed*), it is dropped. If the trigger arrives while the image is still being read out of the sensor, the start of exposure is delayed until the next opportunity to clear the imaging area without injecting noise into the output image. The end of exposure cannot occur before the end of the previous image readout. Therefore, exposure start may be delayed to ensure this, which means priority is given to maintaining the proper exposure time instead of to the trigger start.



Trigger Mode 14 ("Overlapped Exposure/Readout Mode")

Registers—TRIGGER_MODE: 830h		
Presence	[0]	1
ON	[6]	1
Polarity	[7]	Low/High
Source	[8-10]	GPIO Pin
Value	[11]	Low/High
Mode	[12-15]	Trigger_Mode_14
Parameter	[20-31]	None

3.7 Multi-Shot Trigger (Mode 15)

Multi-Shot mode is a vendor-unique trigger mode that allows the user to fire a single hardware or software trigger and have the camera acquire and stream a predetermined number of images.

The number of images to be acquired is determined by the parameter specified with the trigger mode. This allows up to 255 images to be acquired from a single trigger. Setting the parameter to 0 results in a non-free running, non-overlap mode. If Trigger Mode 14 is supported by the camera, setting the parameter to 0 results in a non-free running, overlap mode.

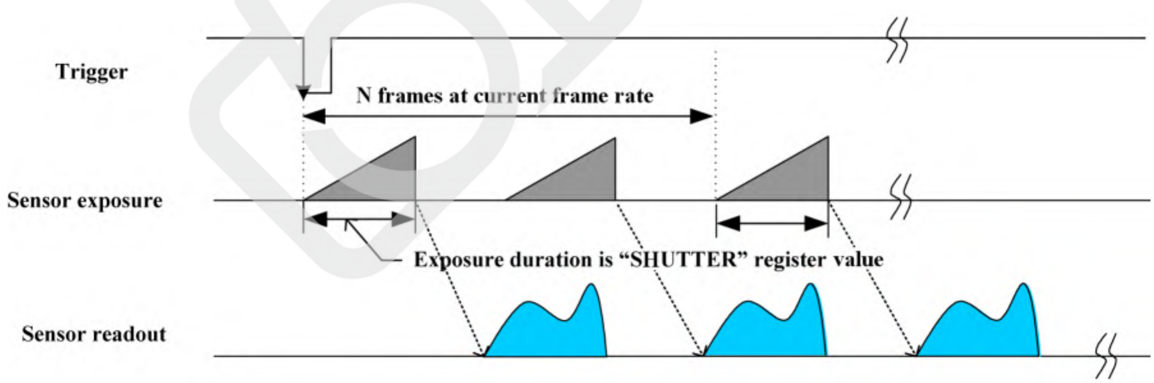
Once the trigger is fired, the camera will acquire *N* images with an exposure time equal to the value defined by the shutter, and stream the images to the host system at the current frame rate. Once this is complete, the camera can be triggered again to repeat the sequence.

Any changes to the trigger control cause the current sequence to stop.

> **Note:** During the capture of *N* images, provided that N>0, the camera is still in an asynchronous trigger mode (essentially Trigger Mode 14), rather than continuous (free-running) mode. The result of this is that the frame rate is turned OFF, and the camera put into extended shutter mode. Users should ensure that the maximum shutter time is limited to 1/frame_rate to get the *N* images captured at the current frame rate.

Related Knowledge Base Articles

Title
Extended shutter mode operation for FLIR machine vision cameras



Trigger Mode 15, when N is greater than 0. ("Multi-Shot Trigger Mode")

Registers—TRIGGER_MODE: 830h		
Presence	[0]	1
ON	[6]	1
Polarity	[7]	Low/High
Source	[8-10]	GPIO Pin
Value	[11]	Low/High
Mode	[12-15]	Trigger_Mode_15
Parameter	[20-31]	*N* number of images to be acquired

3.8 Working with GPS Data

You can use a GPS receiver in conjunction with a Ladybug camera to record GPS data with stream files, generate Google Map or Google Earth files, and download a GPS data file.

> **Note:** You can record images to stream files when the GPS location changes after a specified distance. This feature is available using the Ladybug API. For more information, see the ladybugSimpleRecording example.

When using a GPS receiver with your Ladybug, keep in mind the following:

- Your GPS receiver should have a serial or USB interface for connecting with your laptop and be able to stream NMEA 0183 data in real time.
- To provide reliable data, your GPS device should show a connection with at least 3 satellites.
- It may take some time between when you first connect the GPS device to your PC and when it is recognized and configured for use with LadybugCapPro.
- The following GPS NMEA data structures are supported: GPGGA, GPGSA, GPGSV, GPRMC, GPZDA, GPVTG and GPGLL.

For information about how GPS data is incorporated into stream files, see Stream File Format.

Configuring the GPS receiver

Before capturing GPS data, use the LadybugCapPro Options button () on the Main Toolbar to specify some basic settings for communicating with your GPS receiver.

Control	Description
Port Number	The port to which the GPS receiver is connected. To determine the port, expand the Ports node in the Windows Device Manager. **Note:** LadybugCapPro does not automatically detect this setting upon startup.
Baud Rate	The signaling event rate at which the GPS receiver communicates with the PC. This rate is limited by what the GPS unit supports. The NMEA 0183 standard supports the default value of 4800.
Data Update Interval	The time interval at which positional data is updated from the GPS to the PC. This rate can be set up to the maximum supported by the GPS unit. The default value is 1000 ms.
Start GPS when starting LadybugCapPro	When checked, specifies that the GPS unit should transmit positional data as soon as the LadybugCapPro application starts in live camera mode, using the existing settings.
Google Map Height /Google Map Width	Specifies the dimensions of the Google Maps that are generated. These dimensions affect the amount of area covered in the maps, rather than their resolution.

Using the GPS Toolbar

Once you have configured your GPS receiver, you are ready to use the GPS toolbar to record GPS data and generate Google Map or Google Earth files.

Icon	Description
	Instructs LadybugCapPro to begin receiving positional data from the GPS unit. When used in conjunction with Capturing Stream Files, GPS data is saved with the stream file. For more information, see Stream File Format. This control is not available in recorded stream mode. Click again to stop GPS recording.
	Creates a Google Map file from the GPS data that was previously recorded with the stream file, and allows you the option to load it. An internet connection is required to view the file. Google Maps are saved as .html files in the bin folder of the PGR Ladybug installation directory. This control is not available in live image-grabbing mode.
	Creates a Google Earth file from the GPS data recorded with the stream file, and allows you the option to load it. The Google Earth application and an internet connection are required to view the file. Google Earth files are stored as .kml files in the bin folder of the PGR Ladybug installation directory. This control is not available in image capture mode.

Generating a GPS data file

You can download the data file containing the GPS data for each frame of a recorded stream file. From the **GPS** menu item, select **Generate GPS/frame information**. After the file is generated, a dialog box informs you of the location of the file.

3.8.1 Using GPS with the Ladybug API

For a code example, please see the ladybugSimpleGPS example. Examples can be accessed from:

Start Menu -> Teledyne Ladybug SDK-> Examples

The Ladybug library has the ability to interface with a GPS device and insert NMEA sentence data into Ladybug images. The data can then be extracted at a later time and be used to generate HTML data, which can be displayed as a Google Map, or KML data, which can be loaded into Google Earth.

The NMEA sentences supported by the Ladybug library are:

- GPGGA
- GPGSA
- GPGSV
- GPRMC
- GPZDA
- GPVTG
- GPGLL

Detecting the GPS COM Port

Using the GPS functionality requires the use of a GPS device. The COM port that the GPS device is connected to must be known. To determine the port, perform the following steps:

- Right click on "My Computer".
- Click on the Hardware tab and click the "Device Manager" button.
- Expand the "Ports (COM & LPT)" node and note the COM port that the GPS device is mapped to.

Using the Ladybug API for GPS

The following steps provide a brief overview of how to use the GPS functionality of the Ladybug library:

1. Create a GPS context (`LadybugGPSContext`) by calling `ladybugCreateGPSContext()`. This may be done at the same time as the creation of the Ladybug camera context.
2. Register the GPS context with the Ladybug camera context by calling `ladybugRegisterGPS()`. A single GPS context can be registered with several Ladybug camera contexts.
3. Initialize the device by calling `ladybugInitializeGPS()`.
4. Start the GPS device by calling `ladybugStartGPS()`. This may be called when `ladybugStart()` is called. It takes about 5 seconds for the GPS data to become available.
5. Once image grabbing is active, there are several options for image grabbing. The options, with further explanations below, are:

   - ### *Getting NMEA data from a GPS device or LadybugImage*

     The functions `ladybugGetGPSNMEAData` or `ladybugGetGPSNMEADataFromImage` can be used to get a single NMEA sentence from a GPS device or `LadybugImage`. This is usually sufficient if only a small set of values are needed (for example, only latitude and longitude).

If all the sentences are required, calling `ladybugGetAllGPSNMEAData` or `ladybugGetAllGPSNMEADataFromImage` will populate a `LadybugNMEAGPSData` structure with all the supported NMEA sentences (if available).

Each NMEA structure has a boolean value called `bValidData`. This value is true only if the data contained in that structure is valid.

- *Getting GPS data from a LadybugImageInfo structure*

  When grabbing images in JPEG mode, a filled `LadybugImageInfo` structure is available in each `LadybugImage`. When the GPS functionality is active, the following values are populated:

  - dGPSAltitude
  - dGPSLatitude
  - dGPSLongitude

  If any of these values are equal to `LADYBUG_INVALID_GPS_DATA`, then they should be considered invalid.

6. Once image grabbing has been completed, call `ladybugStopGPS()` to stop data acquisition from the GPS device.
7. Unregister the GPS context by calling `ladybugUnregisterGPS()`.
8. Destroy the context by calling `ladybugDestroyGPSContext()`.

3.8.2 Generating Google Maps and Google Earth data

The Ladybug library allows the user to retrieve GPS data from a stream file and automatically generate Google Maps or Google Earth data, which can then be loaded in their respective applications.

Using LadybugCapPro:

From the GPS menu, select Generate Google Map HTML or click the  button.

From the GPS menu, select Generate Google Earth KML or click the  button.

Using Ladybug API:

If a stream context has already been initialized for reading, calling `ladybugWriteGPSSummaryDataToFile` with the relevant `LadybugGPSFileType` generates GPS data for the entire stream file.

4 Ladybug Attributes

4.1 Brightness

Brightness, also known as offset or black level, controls the level of black in an image.

The camera supports brightness control.

To adjust brightness:

- **During Capture**—From the Settings menu, select Camera Control and click the Camera Settings tab.
- **During Post Processing**—From the Settings menu, select Image Processing and select Black Level to make an adjustment with the slider.
- CSRs—Imaging Parameters: 800h-888h

4.2 Shutter Time

The Ladybug supports Automatic, Manual, and One Push control of the image sensor shutter time.

> **Note:** The terms “integration”, “exposure” and "shutter" are interchangeable.

Shutter times are scaled by the divider of the basic frame rate. For example, dividing the frame rate by two (e.g. 15 FPS to 7.5 FPS) causes the maximum shutter time to double (e.g. 66 ms to 133 ms).

The maximum shutter time can be extended beyond the normal range by disabling the frame rate. Once the frame rate is disabled, you should see the maximum value of the shutter time increase.

The supported shutter time range is:

The time between the end of shutter for consecutive frames is always constant. However, if the shutter time is continually changing (e.g. being controlled by Auto Exposure), the time between the beginning of consecutive integrations will change. If the shutter time is constant, the time between integrations will also be constant.

The camera continually exposes and reads image data off of the sensor under the following conditions:

1. The camera is powered up; and
2. The camera is in free running, not asynchronous trigger, mode. When in trigger mode, the camera simply clears the sensor and does not read the data off the sensor.

The camera continues to expose images even when data transfer is disabled and images are not being streamed to the computer. The camera continues exposing images in order to keep things such as the auto exposure algorithm (if enabled) running. This ensures that when a user starts requesting images, the first image received is properly exposed.

When operating in free-running mode, changes to the shutter value take effect with the next captured image, or the one after next. Changes to shutter in asynchronous trigger mode generally take effect on the next trigger.

To adjust shutter:

- **During Capture**—From the Settings menu, select Camera Control and click the Camera Settings tab.
- CSRs—Imaging Parameters: 800h-888h

To enable extended shutter:

- **During Capture**—From the Settings menu, select Camera Control and click the Camera Settings tab. Deselect Frame Rate On/Off to disable the frame rate.

4.2.1 Shutter Range

The camera offers three preset shutter range modes to set the maximum shutter value:

- Drive Highway—maximum shutter is set to as short as possible to prevent motion blur. Best used outdoors or images may be too dark. This is the default.
- Drive City—maximum shutter set for city driving speed (40 km/h or 25 m/h) to prevent motion blur.
- Indoor—maximum shutter is slightly longer than the driving modes, for use in indoor or lower light applications.
- Low Noise—maximum shutter range allowed for lowest noise. This may introduce motion blur.

To set the shutter range:

- **During Capture**—From the Live Camera Toolbar, make a selection from the Shutter range drop-down.

4.3 Gain

Gain is the amount of amplification that is applied to a pixel by the A/D converter. An increase in gain can result in a brighter image but also an increase in noise.

The Ladybug supports Automatic and One Push gain modes. The A/D converter provides a PxGA gain stage (white balance/preamp) and VGA gain stage. The main VGA gain stage is available to the user, and is variable between models per the table below.

Camera	Gain Range
Ladybug6	0 - 18 dB
Ladybug5+	0 - 18 dB
Ladybug3	-2.25 dB to 24 dB in 0.04 dB increments

To adjust gain:

- **During Capture**—From the Settings menu, select Camera Control and click the Camera Settings tab.
- **During Post Processing**—From the Settings menu, select Image Processing. Select Exposure and then select Manual from the drop-down to adjust the Gain with the slider.
- CSRs—Imaging Parameters: 800h-888h

4.4 Auto Exposure

Auto exposure allows the camera to automatically control shutter and/or gain in order to achieve a specific average image intensity. Additionally, users can specify the range of allowed values used by the auto-exposure algorithm by setting the auto exposure range, the auto shutter range, and the auto gain range.

Auto Exposure allows the user to control the camera system's automatic exposure algorithm. It has three useful states:

State	Description
Off	Control of the exposure is achieved via setting both Shutter and Gain. This mode is achieved by setting Auto Exposure to Off, or by setting Shutter and Gain to Manual.
Manual Exposure Control	The camera automatically modifies Shutter and Gain to try to match the average image intensity to the Auto Exposure value. This mode is achieved by setting Auto Exposure to Manual and either/both of Shutter and Gain to Automatic.
Auto Exposure Control	The camera automatically modifies the value in order to produce an image that is visually pleasing. This mode is achieved by setting the all three of Auto Exposure, Shutter, and Gain to Automatic. In this mode, the value reflects the average image intensity.

Auto Exposure can only control the exposure when Shutter and/or Gain are set to Automatic. If only one of the settings is in "auto" mode then the auto exposure controller attempts to control the image intensity using just that one setting. If both of these settings are in "auto" mode the auto exposure controller uses a shutter-before-gain heuristic to try and maximize the signal-to-noise ratio by favoring a longer shutter time over a larger gain value.

In absolute mode, an exposure value (EV) of 0 indicates the average intensity of the image is 18% of 1023 (18% gray).

The auto exposure algorithm is only applied to the active region of interest, and not the entire array of active pixels.

There are four parameters that affect Auto Exposure:

Auto Exposure Range—Allows the user to specify the range of allowed exposure values to be used by the automatic exposure controller when in auto mode.

Auto Shutter Range—Allows the user to specify the range of shutter values to be used by the automatic exposure controller which is generally some subset of the entire shutter range.

Auto Gain Range—Allows the user to specify the range of gain values to be used by the automatic exposure controller which is generally some subset of the entire gain range.

Auto Exposure ROI—Allows the user to specify a region of interest within the full image to be used for both auto exposure and white balance. The ROI position and size are relative to the transmitted image. If the request ROI is of zero width or height, the entire image is used.

Auto exposure can be controlled on each of the six sensors independently.

To control auto exposure:

- **During Capture**—From the Settings menu, select Camera Control and click the Camera Settings tab.
- **During Post Processing**—From the Settings menu, select Image Processing. Select Exposure and then select Automatic from the drop-down to adjust with the slider. Select an ROI from the drop-down. See .
- CSRs—AUTO_EXPOSURE: 804h and AE_ROI: 1A70 – 1A74h

4.4.1 Auto Exposure ROI

There are three preset modes for the auto exposure algorithm:

- Bottom 50%—uses only the bottom 50% of the five side cameras and excludes the top camera from its calculations.
- Top 50%—uses only top 50% of the five side cameras and includes the top camera in its calculations. This is the upside down version of the first mode, used when the camera is mounted upside down (for example, on a helicopter).
- Full Image—uses the entire image of all six cameras for its calculations. This is the default.

For some data formats – 8-bit and JPEG12Processed – the auto exposure modes are set for image capture using LadybugCapPro and set for post processing using the Ladybug API.

For other data formats – Raw 12-bit and 16-bit – the auto exposure modes are set both for image capture and post processing using LadybugCapPro.

To select an auto exposure ROI:

During Capture:

- From the Live Camera Toolbar, make a selection from the AE ROI drop-down.

During Post Processing:

1. From the Settings menu, select Image Processing.
2. Select Exposure and then select Automatic from the drop-down to adjust with the slider.
3. Select an ROI from the drop-down.

4.5 Independent Sensor Control of Shutter, Gain and Auto Exposure

The Independent Sensor Control feature provides customized control of exposure for each of the six cameras on the camera system independently. This feature allows users to acquire images with greater dynamic range of the overall scene.

> **Note:** Independent Sensor Control provides independent control of exposure-related features only. This feature does not encompass other camera control settings such as gamma or white balance.

Global control applies the same setting (automatic or manual) and value to all six cameras. Independent control allows separate settings (automatic or manual) and values to each camera.

- If global shutter is off, shutter can be independently controlled.
- If global gain is off, gain can be independently controlled.
- If either global shutter or global gain is off, exposure can be independently controlled.

To control shutter, gain, and exposure:

During Capture:

- From the Settings menu, select Camera Control, or click the button.
    - Global control of shutter, gain, and exposure is set on the Camera Settings tab.
    - Independent control of shutter, gain, and exposure is set on the Ladybug Settings tab.

Using Registers:

- Use the shutter and gain registers to turn off global control of Shutter and Gain.
    - For details of these registers see SHUTTER: 81Ch and GAIN: 820h
- Use SUB_AUTO_EXPOSURE_* to independently control the exposure for each camera.
- Use SUB_SHUTTER_* and SUB_GAIN_* to independently control shutter and gain for each camera.
    - For details of these registers see INDEPENDENT_CONTROL_INQ: 1E94h.

4.6 Gamma and Lookup Table

The camera supports gamma functionality.

Sensor manufacturers strive to make the transfer characteristics of sensors inherently linear, which means that as the number of photons hitting the imaging sensor increases, the resulting image intensity increases are linear. Gamma can be used to apply a non-linear mapping of the images produced by the camera. Gamma is applied after analog-to-digital conversion and is available in all data formats except Raw. Gamma values between 0.5 and 1 result in decreased brightness effect, while values between 1 and 4 produce an increased brightness effect. By default, Gamma is enabled and has a value of 1.25. To obtain a linear response, disable gamma.

For 8-bit, gamma is applied as:

```
OUT = 255*(IN/255)^(1/gamma)
```

Related Knowledge Base Articles

Title
How is gamma calculated and applied?

To adjust gamma:

- **During Capture**—From the Settings menu, select Camera Control and click the Camera Settings tab.
- **During Post Processing**—From the Settings menu, select Image Processing. Select Gamma to adjust with the slider. See Adjusting Unprocessed Images.

4.7 White Balance

The Ladybug supports white balance adjustment, which is a system of color correction to account for differing lighting conditions. Adjusting white balance by modifying the relative gain of R, G and B in an image enables white areas to look "whiter". Taking some subset of the target image and looking at the relative red to green and blue to green response, the objective is to scale the red and blue channels so that the response is 1:1:1.

The user can adjust the red and blue values. Both values specify relative gain, with a value that is half the maximum value being a relative gain of zero.

White Balance has two states:

State	Description
Off	The same gain is applied to all pixels in the Bayer tiling.
On/Manual	The Red value is applied to the red pixels of the Bayer tiling and the Blue value is applied to the blue pixels of the Bayer tiling.

The following table illustrates the default gain settings for most cameras.

	Red	Blue
Black and White	32	32
Color	1023	1023

The camera can also implement Automatic and One Push white balance. One use of Automatic and One Push white balance is to obtain a similar color balance between cameras that are slightly different from each other. In theory, if different cameras are pointed at the same scene, using Automatic and One Push results in a similar color balance between the cameras.

One Push only attempts to automatically adjust white balance for a set period of time before stopping. It uses a “white detection” algorithm that looks for “whitish” pixels in the raw Bayer image data. One Push adjusts the white balance for a specific number of iterations; if it cannot locate any whitish pixels, it will gradually look at the whitest objects in the scene and try to work off them. It will continue this until has completed its finite set of iterations.

Automatic is continually adjusting white balance. It differs from One Push in that it works almost solely off the whitest objects in the scene.

Note: The white balance of the camera before using Automatic and One Push must already be relatively close; that is, if Red is set to 0 and Blue is at maximum (two extremes), Automatic and One Push will not function as expected. However, if the camera is already close to being color balanced, then Automatic and One Push will function properly.

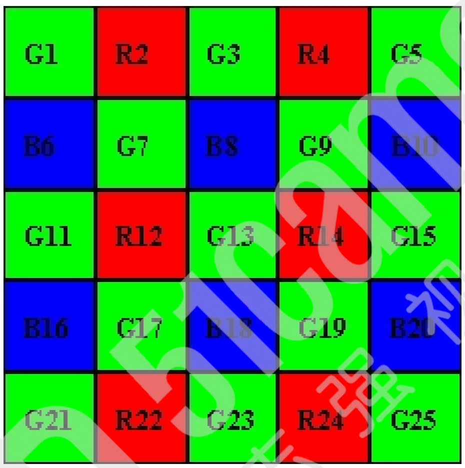
To adjust white balance:

- **During Capture**—From the Settings menu, select Camera Control and click the Camera Settings tab.
- **During Post Processing**—From the Settings menu, select Image Processing. Select White Balance to make custom adjustments with the sliders, or select presets from the drop-down list.
- CSRs— WHITE_BALANCE: 80Ch

4.8 Color Processing

A Bayer tile pattern color filter array captures the intensity red, green or blue in each pixel on the sensor. The image below is an example of a Bayer tile pattern.

To determine the actual pattern on your camera, query BAYER_TILE_MAPPING: 1040h.



Example Bayer Tile Pattern

To control color processing:

- **During Post Processing**—Click the button to select the algorithm used to convert raw Bayer-tiled image data to 24-bit RGB images. Lower-quality algorithms can increase the LadybugCapPro display rate, and higher-quality algorithms can decrease the display rate.

  Two additional algorithms are:

  - **High Quality Linear on GPU**: Same output as High Quality Linear, but better performance on graphics cards with NVidia CUDA support.
  - **Directional Filter**: Highest quality output, but significantly better performance than Rigorous.
- CSRs—BAYER_TILE_MAPPING: 1040h

Accessing Raw Bayer Data

Users interested in accessing the raw Bayer data to apply their own color conversion algorithm or one of the SDK library algorithms should acquire images using a video mode that supports Raw pixel encoding.

The actual physical arrangement of the red, green and blue "pixels" for a given camera is determined by the arrangement of the color filter array on the imaging sensor itself. The format, or order, in which this raw color data is streamed out, however, depends on the specific camera model and firmware version.

Raw image data can be accessed programmatically via the pData pointer in the LadybugImage structure (e.g. LadybugImage.pData). In Raw8 modes, the first byte represents the pixel at (row 0, column 0), the second byte at (row 0, column 1), etc. In the case of a camera that is streaming Raw8 image data in RGGB format, if we access the image data via the pData pointer we have the following:

- pData[0] = Row 0, Column 0 = red pixel (R)
- pData[1] = Row 0, Column 1 = green pixel (G)
- pData[1616] = Row 1, Column 0 = green pixel (G)
- pData[1617] = Row 1, Column 1 = blue pixel (B)

Related Knowledge Base Articles

Title
Writing color processing software and color interpolation algorithms
How is color processing performed on my camera's images?

4.9 High Dynamic Range (HDR) Imaging

Generally speaking, digital camera systems are not capable of accurately capturing many of the high dynamic range scenes that they are exposed to in real world settings. That is, they may not be able to capture features in both the darkest and brightest areas of an image simultaneously - darker portions of the image are too dark or brighter portions of the image are too bright. High Dynamic Range (HDR) mode helps to overcome this problem by capturing images with varying exposure settings. HDR is best suited for stationary applications.

The camera can be set into an HDR mode in which it cycles between 4 user-defined shutter and gain settings, applying one gain and shutter value pair per frame. This allows images representing a wide range of shutter and gain settings to be collected in a short time to be combined into a final HDR image later. The camera does not create the final HDR image; this must be done by the user.

The HDR interface contains gain and shutter controls for 4 consecutive frames. When **Enable high dynamic range** is checked, the camera cycles between settings 1-4, one set of settings per consecutive frame.

To enable HDR:

- **During Capture**—From the Settings menu, select Camera Control and click the High Dynamic Range tab.
- **Ladybug SDK example program**—use the ladybugCapterHDRImage example
- CSRs—HDR: 1800h – 1884h

Related Knowledge Base Articles

Title
Capturing HDR Images with Ladybug cameras

4.10 Ladybug Image Information

Ladybug image information is stored at the beginning of each image and contains frame-specific data. This data is separate from the image payload.

The following frame-specific information is provided:

- Timestamp in seconds and microseconds
- Sequence number of the image
- Horizontal refresh rate
- Gain used by each of the 6 cameras
- Shutter for each of the 6 sensors
- Brightness
- White Balance
- Gamma
- Bayer Gain
- Bayer Map
- Serial number of the camera

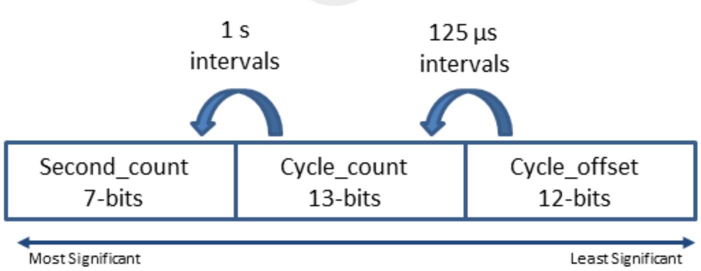
> **Note:** Ladybug image values are those in effect at the end of shutter integration.

The LadybugImageInfo structure in ladybug.h contains the format of the embedded information. This information is always provided at the start of the image.

Interpreting Timestamp information

The CYCLE_TIME register is located at 1EA8h.

The Timestamp format is as follows:



Cycle_offset increments from 0 to *x* depending on implementation, where *x* equals one cycle_count.

Cycle_count increments from 0 to 7999, which equals one second.

Second_count increments from 0 to 127.

All counters reset to 0 at the end of each cycle.

Note: On USB3 devices, the four least significant bits of the timestamp do not accurately reflect the cycle_offset and should be discounted.

5 Post Processing Control

> **Note:** The options available for post processing control are dependent on the data format chosen during image capture.

Certain parameters can be adjusted after image capture during post processing, including:

- Stabilization
- Vertical tilt
- Stitching
- Image parameters such as black level, exposure, gamma, tone mapping, white balance

For 8-bit and JPEG12Processed formats see Adjusting 8-bit and JPEG12Processed Images.

For 12-bit and 16-bit formats see Adjusting Unprocessed Images.

5.1 Reading Stream Files

Using LadybugCapPro:

From the File menu, select New or click the  button. Click Load Stream File. Select your file and click Open.

> **Note:** When LadybugCapPro is launched, it prompts you to start a camera or load a stream file.

Using Ladybug API:

The following steps provide a brief overview of how to use the stream functionality of the Ladybug library to read a stream from disk:

1. Create a stream context (`LadybugStreamContext`) by calling `ladybugCreateStreamContext()`.
2. Initialize the stream context for reading by calling `ladybugInitializeStreamForReading()`.
3. At this point, additional information about the stream can be obtained by calling `ladybugGetStreamHeader` and `ladybugGetStreamNumOfImages`.
4. If a specific image is required, calling `ladybugGoToImage()` will move the stream to the specified image. Otherwise, `ladybugReadImageFromStream()` will retrieve the image from the current reading pointer.
5. When reading is complete, call `ladybugStopStream()` to stop reading.

6. Destroy the context by calling `ladybugDestroyStreamContext()` when suitable (such as program termination).

5.2 Working with Images

In both live camera and recorded video modes, you can use the Image Processing Toolbar to change the way the camera processes and renders images. You can also click and drag inside the image display to render image rotation and magnification in different ways.

5.2.1 Falloff Correction

Falloff Correction adjusts the intensity of light in images to compensate for a vignetting effect. This control is disabled by default.

To enable fall off correction:

1. From the Imaging Processing toolbar click the button, or, from the Settings menu select Falloff Correction.
2. Select Enable falloff correction.
3. Specify an attenuation value with the slider or textbox. The attenuation value regulates the degree of adjustment to apply.
4. Click OK.

5.2.2 Blending Width

Blending is the process of adjusting pixel values in each image that overlap with the fields of adjacent images to minimize the effect of pronounced borders. The default width of 100 pixels is suitable for the 20-meter sphere radius to which Ladybug cameras are pre-calibrated. To change the sphere radius see Adjusting Sphere Size for Stitching.

The blending width control allows you to adjust the pixel width along the sides of each of the six images within which blending takes place prior to stitching.

To modify the blending width:

1. From the Imaging Processing toolbar click the button, or, from the Settings menu select Blending Width.
2. Specify a blending width with the slider or textbox.
3. Click OK.

5.2.3 Rendering the Image for Display

> **Note:** These controls affect video display only. To specify how images are rendered when outputting to video, specify an Output

> Type using the Stream Processing Toolbar.

You can change the way images are rendered for display. See Projection Types for detailed information.

To modify the image display:

1. From the Imaging Processing toolbar click the button, then select an image type from the drop-down list.

Display	Description
Panoramic	Renders the image as a panoramic projection. This is the default display. Use **Mapping Type** to specify either a radial or cylindrical projection.
Spherical 3D	Renders the image as a 3-dimensional spherical projection.
Dome Projection	Renders the image as a dome projection.
All-Camera	Six images from each camera are rendered separately, unstitched.
Single-Camera (Raw)	Image data from a selected camera is displayed.
Single-Camera (Rectified)	Image data from a selected camera is displayed and rectified to account for lens distortion. Rectification is the process of generating an image that fits a pin-hole camera model.

2. For the Rotation Angle, from the Imaging Processing toolbar click the button, then select from the drop-down list. The rotation angle specifies the orientation of the camera unit's six cameras to the projection. The default orientation is camera 0 projects to the front of the sphere and camera 5 to the upward pole (or top) of the sphere.

3. For the Mapping Type, from the Imaging Processing toolbar click the button, then select radial or cylindrical from the drop-down list. The mapping projection dictates how the six individual pictures from each camera are stitched into a panoramic display.

Using the Mouse

You can click and drag inside the image display to control the way images render on the screen. These controls do not affect how images are recorded or how the streams are output to other formats. Not all controls work in all display renderings (panoramic, spherical, dome).

Mouse Control	Description
Left-click, drag	Rotates the yaw view in the direction of the drag. In spherical view, the rotation is sustained at a rate proportional to the speed of the drag. Click again to stop rotation.
Right-click, drag horizontally	Rotates the image pitch. For best results, magnify the image.

Mouse Control	Description
Right-click, drag vertically	Rotates the image roll. For best results, magnify the image.
Scroll wheel	Magnifies the image display.
Right-click, drag horizontally + Shift button	Rotates the yaw view horizontally when image is magnified.
Changing window size	Images stretch to fit the current size of the display window.

5.2.4 Stabilizing Image Display

You can adjust the display of images to compensate for the effect of unwanted movement across frames when the camera records on an unstable surface. Image stabilization can be enabled in both live camera and recorded video modes.

Image stabilization is purely image-based. It does not use external sensors to detect motion. Instead, it compares image patterns across successive frames. Therefore, in order for image stabilization to produce good results, the following requirements should be met:

- Shutter speed must be fast enough to produce clear images without motion blur. Images produced outdoors during daylight hours should not be a concern. In darker places, you may need to set the shutter speed manually.
- There must be patterns across images. If entire images only contain simple textures, such as clear sky or a white wall, the algorithm will have difficuly finding patterns. It is not necessary for all the cameras in the system to have patterns; having patterns on some cameras may suffice. Additionally, patterns should be distant. If they are too close to the camera, there may be errors.
- The frame rate should be fast enough so that the relative movement across frames is not large. If the frame rate is low and the relative movement of images across frames is large, the algorithm may be unable to find patterns. The faster the movement, the faster the frame rate should be.

Although you can enable image stabilization during both live camera and recorded video modes, we recommend using it primarily when outputting stream files. The stabilization algorithm is resource-intensive. When stabilization is enabled during live camera mode, the system may be unable to perform the necessary computations while keeping up with all the incoming frames.

To enable Image Stabilization:

- In LadybugCapPro, select **Enable Stabilization** from the **Settings** menu, or click the Stabilization ( ) icon on the Image Processing toolbar. For more information, see Working with Images.
- Using the Ladybug API, invoke the ladybugEnableImageStabilization function. For more information, refer to the **LadybugProcessStream** example, available from the Windows **Start** menu -> **Teledyne Ladybug SDK** -> **Examples**.

Once enabled, the panoramic, spherical and dome view outputs become stabilized. Additionally, stream files that are output to JPEG, bmp, or avi files are stabilized.

To set Stabilization Parameters:

Depending on your requirements, you can adjust the following image stabilization parameters:

Decay Rate - Specifies the degree of correction across images that the stabilization algorithm uses. The default setting is 0.9. A setting of 1.0 (maximum) instructs the algorithm to apply correction across all positional difference. A decay rate of 1.0 may result in an undesired drift effect, as the camera's original position may shift slowly over a long period of time. Other unwanted effects may result. For example, when capturing images from a car as it turns at intersections, the algorithm will attempt to recover the image pattern produced before the turn. By setting this value a little lower than 1.0, the display slowly re-adjusts, cancelling the drift effect.

Maximum Search Range - The stabilization algorithm searches for patterns within a series of templates in each frame. This value specifies the size, in pixels, of each template. If the frame rate is low or movement is fast, try using a larger value for better results. However, keep in mind that a larger value requires more computation.

To Set stabilization parameters, select **Options** from the **Settings** menu, or click the Options ( ) icon on the Main toolbar.

5.2.5 Adjusting Sphere Size for Stitching

Using the sphere size control ( ) on the Image Processing toolbar, you can minimize parallax by changing the sphere radius to which images are calibrated for stitching panoramas. The following options are available:

Fixed Size

By default, the sphere radius is calibrated at 20 m, which is well-suited for most outdoor scenes. However, if most subjects in the scene are closer than 20 m, you may get better stitching results by choosing a smaller radius. A larger sphere radius of 100 m is also available for scenes that are more distant.

Selecting a fixed size disables dynamic stitching.

One-Shot Dynamic Stitch

One-shot dynamic stitch calculates an optimal sphere radius for the entire scene. Successive frames are stitched to the same radius until another adjustment is made.

Auto Dynamic Stitch

Auto dynamic stitch calculates optimal stitching distances for different areas of the image, so that these distances vary across the entire image. Stitching distances are re-calculated for each frame. Auto dynamic stitch is best used when distances in the same image coordinates vary greatly across successive frames, and prominent stitching errors cannot be fixed using another stitching calculation technique listed above.

> **Note:** Although you can enable auto dynamic stitch during both live camera and recorded video modes, we recommend using it primarily when outputting stream files. The dynamic stitch algorithm is resource-intensive. When enabled during live camera mode, the system may be unable to perform the necessary computations while keeping up with all the incoming frames.

You can modify the Minimum distance, Maximum distance, and Default distance used for dynamic stitching. From the Settings menu, select Options to open the LadybugCapPro Options dialog.

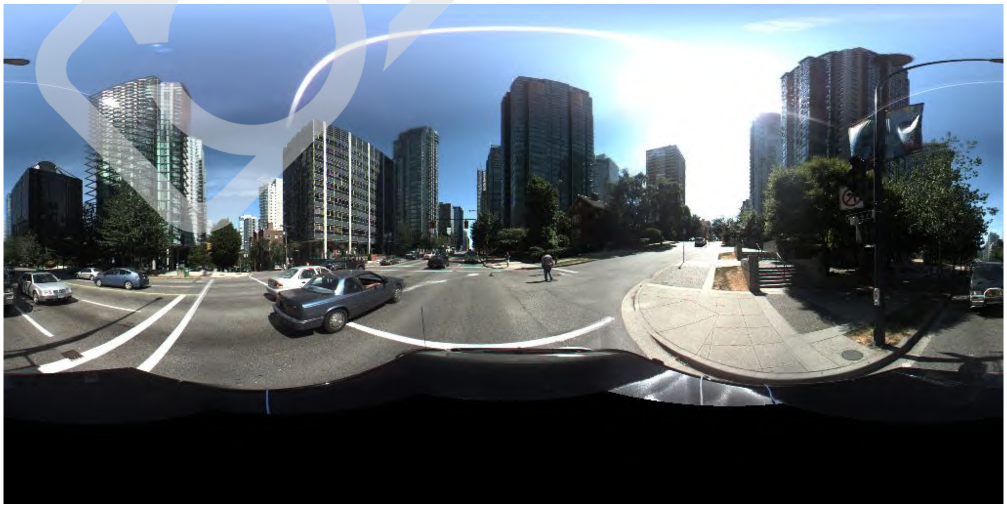
Selecting a fixed size from the drop-down as above disables dynamic stitching.

5.2.6 Adjusting Vertical Tilt

Panoramic images produced by the Ladybug camera system are sensitive to the position of the camera during image capture. Due to the nature of the panoramic mapping, if the camera is at a slight tilt, vertical lines in the scene will appear curved in the image. Since it is not always possible to ensure that the camera is perfectly aligned with the vertical axis in the scene, the Ladybug SDK allows you to correct vertical tilt in images. This is done by selecting **Set Z Axis** in the **Settings** menu in the LadybugCapPro Main Window. You can adjust vertical tilt either during image capture or stream file playback.

The following images show the effect of adjusting for vertical tilt. The image on top shows vertical lines curved. This curvature is corrected in the bottom image.

Scene with Tilt Effect



Tilt-Adjusted Scene

To adjust Vertical Tilt

When you select **Set Z Axis** from the **Settings** Menu, LadybugCapPro prompts you to Shift-click on four points in the image. The first two clicks specify points on a line in the image that should be adjusted vertically, and the second two clicks specify another line in the image that should be adjusted vertically. By doing this, LadybugCapPro can orient the selected lines with the center of the sphere, and re-adjust the Z (vertical) axis of the radial projection. After the fourth click, all the images being captured or replayed are adjusted vertically.

For example, in the first figure above, you might select the following four points to adjust:

Suggested Clicking Points to Specify Vertical Line Adjustment

When selecting vertical line clicking points, keep in mind the following:

- The two points of each line should be as distant as possible.
- The two lines should not be too close to each other, or too close in an exact opposite direction (that is, 180 degrees apart).

> **Note:** Vertical tilt adjustment can also be accomplished by manually adjusting the image roll using the mouse. To adjust image roll, right-click and drag the mouse vertically inside the image. For best results, magnify the image first using the mouse scroll wheel.

To undo Vertical Tilt

To undo any vertical tilt adjustments, click the Rotation Angle control ( ) on the Image Processing Toolbar, and select **Default**.

5.2.7 Adjusting 8-bit and JPEG12Processed Images

Use the Image Processing control ( ) on the Image Processing Toolbar, to make the following adjustments to 8-bit and JPEG12Processed images:

Color Correction

When enabled, overall image hue, intensity and saturation of images can be adjusted. Red, green and blue can also be adjusted individually, which may help to correct white balance issues during image capture.

> **Note:** Color correction may degrade overall image quality.

Sharpening

When enabled, image textures are sharpened. This effect may be most noticeable along texture edges.

Texture Intensity Adjustment

This control is best used when the camera is operating in an independent sensor control mode, or a stream file is opened that was captured in an independent exposure control mode. When enabled, texture intensities are adjusted to compensate for differences in exposure between individual images. The adjustment process converts integer pixel values to floating point values to achieve higher dynamic range (HDR). For best results, use texture intensity adjustment in combination with one of the following techniques:

- Tone mapping (below)
- Output to .HDR format and process with other software. See Viewing and Outputting Stream Files.

Tone Mapping

When enabled, the dynamic range of images is converted from high (HDR) to low (LDR) to resemble more closely the dynamic range of the human eye. When using OpenGL, the following controls are available:

- **Compression**—Gamma-style adjustment re-maps image values. A higher value yields greater compression in bright areas of the image.
- **Local area**—Determines the size of the area around each pixel that is used to calculate new values as part of the overall compression process.

12-bit and 16-bit images have the option of using CPU for better quality but slower processing.

5.2.8 Adjusting Unprocessed Images

Use the Image Processing control ( ) on the Image Processing Toolbar, to make adjustments to unprocessed images.

Luminance

- **Black Level**—Adjustments to the image Black Level can be enabled and set.

- **Exposure**—Adjustments to the image auto exposure settings include selecting an ROI (Full Image, Top camera only, Bottom only). Selecting Automatic adjusts the six images as a whole while selecting Automatic Independent adjusts each image separately. When in Automatic Exposure, compensation can be applied. When in Manual Exposure, gain settings can be adjusted.

Tonal

Gamma:

Sensor manufacturers strive to make the transfer characteristics of sensors inherently linear, which means that as the number of photons hitting the imaging sensor increases, the resulting image intensity increases are linear. Gamma can be used to apply a non-linear mapping of the images produced by the camera. Gamma is applied after analog-to-digital conversion and is available in all data formats except Raw. Gamma values between 0.5 and 1 result in decreased brightness effect, while values between 1 and 4 produce an increased brightness effect. By default, Gamma is enabled and has a value of 1.25. To obtain a linear response, disable gamma.

Gamma can be included in the auto exposure calculation.

Tone Mapping:

When enabled, the dynamic range of images is converted from high (HDR) to low (LDR) to resemble more closely the dynamic range of the human eye. When using OpenGL, the following controls are available:

- **Compression**—Gamma-style adjustment re-maps image values. A higher value yields greater compression in bright areas of the image.

- **Local area**—Determines the size of the area around each pixel that is used to calculate new values as part of the overall compression process.

12-bit and 16-bit images have the option of using CPU for better quality but slower processing.

Color

White Balance:

The Ladybug supports white balance adjustment, which is a system of color correction to account for differing lighting conditions. Adjusting white balance by modifying the relative gain of R, G and B in an image enables white areas to look "whiter". Taking some subset of the target image and looking at the relative red to green and blue to green response, the objective is to scale the red and blue channels so that the response is 1:1:1.

The user can adjust the red and blue values. Both values specify relative gain, with a value that is half the maximum value being a relative gain of zero.

Use the sliders to make custom adjustments, or select a preset from the drop-down list.

Saturation—use the slider controls to make adjustments to the vibrancy of the image.

Levelling controls:

Set the Black Point Level and White Point Level using the sliders.

Use the Grey eyedropper control to set the white balance by clicking on a region of the image that should be a shade of grey.

Use the White eyedropper control to set the white balance and white point by clicking on a region of the image that should be white.

Miscellaneous

- **Smear Correction**—When enabled, smear is corrected for either unsaturated or full correction.
- **Noise reduction**—When enabled, noise present in the image is reduced.
- **Sharpening**—When enabled, image textures are sharpened. This effect may be most noticeable along texture edges.
- **False color removal**—When enabled, removes rainbow sparkles in the image caused by small points of light.

Adjustments to an image can be saved by clicking the Save to settings button. Saved settings are in use the next time the file is opened. Saved settings can also be used for all images in the stream by clicking the Apply to all images button. To return to the default settings, click the Reset button. To return to a saved configuration, click the Load from settings button.

Saved settings are persistent and unique to a particular file.

5.2.9 Histogram

Displays a histogram of the values represented in the pixels of the current image.

To display a histogram:

1. From the Imaging Processing toolbar click the  button, or, from the Settings menu select Histogram.
2. From Image Information select the channels to view. Red, Green, and Blue are all selected by default.
3. From the Options, select:

- **Max Percent** allows you adjust the graphical display to view a subset of percentage representation. For example, to view only the first 5% of the representation of values in the graph, enter '5' in the Max Percent field.
- **All Cameras** specifies that the values are compiled from all six cameras on the Ladybug system. To see values from only one camera at a time, select a camera. (For camera orientation, see Rendering the Image for Display.)

5.3 Saving Images

In both Live Camera and Stream File mode, you can save the current image to panoramic JPEG or panoramic bitmap format, or as six individual color-processed bitmap images, rectified or non-rectified. Images are saved to the My Documents folder.

To save images:

From the **Image** menu, select one of the following:

- **Save Panoramic JPG** or **Save Panoramic BMP**. You can select a pre-defined resolution, or a custom resolution.
    - When **Custom Size** is selected for the first time, enter your desired dimensions. To change your custom dimensions, select **Change Custom Size**.

- To set the quality of JPEG compression, see 'Main Toolbar' in LadybugCapPro Main Window.
- Saved Images are rendered only in Panoramic display format, regardless of display setting.

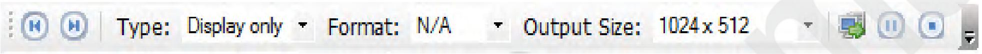
- **Save 6 Color Processed Images BMP** or **Save 6 Rectified Images BMP**. These options allow you save separate color-processed images from each of the six cameras on the system. If the second option is chosen, images are rectified to correct lens distortion.

5.4 Viewing and Outputting Stream Files

When you load a previously-recorded stream file, you can use the the Stream Toolbar to navigate the frames of the stream file and output the file to a variety of different video or image formats.

Use the Stream Navigation Toolbar to navigate to specific frames for the output.

Use the Stream Processing Toolbar to set the output type, format and size, and to convert the file for output.



Selecting an Output Type

A drop-down list of formats for outputting the stream. For more information about these formats, see Projection Types.

Output Type	Description
Display only	Displays the frames successively. This option does not create a new file.
Panoramic	Outputs the stream in panoramic view. All output formats are supported. If JPG, BMP, PNG, or TIFF are chosen, separate files are created for each stitched frame. For more information about .flv output, see Outputting Flash Video. To specify a radial or cylindrical mapping projection, select a Mapping Type using the Image Processing Toolbar.
Dome	Outputs the stream in dome view. Output formats are the same as **Panoramic**, above.
Stream file	Saves the stream file as another .pgr stream file. This option is useful for creating a stream file out of a subset of frames from the original.
6 Processed	Saves each individual image from each of the camera system's six cameras, that comprises the specified output, as separate BMP or TIFF files. Images are not rectified to correct lens distortion.
6 Rectified	Similar to '6 Processed,' except each image is rectified to correct lens distortion. Supports JPG, BMP, PNG, and TIFF formats.

Output Type	Description
6 Cube map	Saves six individual images that can be used to construct a cube covering the entire field of view. Supported formats are JPG, BMP, PNG, and TIFF. Cube mapping is an environment mapping method that is useful for creating video game skyboxes and other computer graphics applications. Cube mapping produces an image that is less distorted than panoramic, cylindrical or dome views. Images are named ladybug_cube_XXXXXX_Y, where XXXXXX is the frame number and Y indicates the cube face, 0-5. Cube face is numbered as follows: 0-front 1-right 2-back 3-left 4-top 5-bottom

To convert Stream Files

1. Click the button to start conversion. The Confirm Settings dialog opens for specifying an output directory for the output file.
2. If outputting to AVI format, you can specify a video encoding codec compressor. The compressors that are listed depend on the compression software currently installed on your system. Compressors that we have tested and are known to work correctly are shaded a different color. For more information about recommended codecs, see Recommended codecs for AVI output in Ladybug SDK.
3. Parallel processing—When selected, instructs LadybugCapPro to create multiple threads to speed up image processing. This consumes additional system resources. For best performance, we recommend the following system configuration:
   - Multi-core CPU
   - 16 GB RAM or more
   - Optimized hard disk drive configuration, such as RAID 0
   - Multiple hard drives or partitions, to write the video stream to a different drive.
4. After specifying all applicable settings, click **Convert!** to create the output file(s).

6 About Stream Files

A Ladybug Stream is a set of files that are used to store Ladybug images. A Ladybug Stream consists of one or more Ladybug Stream Files that have the same *Stream base name*. Each stream file contains one or more Ladybug images. Depending on the camera configuration, a Ladybug image in stream file consists of either compressed JPEG image data or uncompressed raw image data. Ladybug images may be written to or extracted from the stream files by using a set of Ladybug API functions defined in `ladybugstream.h`.

6.1 Stream File Usage

For information on how to use stream files with your application, see How-To: Using Stream Files.

For information on using stream files with the LadybugCapPro application, see Capturing Stream Files.

6.2 Stream File Format

For information about the naming convention, format and data structure of stream files, see Stream File Format.

6.3 Stream Files

Ladybug images are written to a set of Ladybug stream files. The size of each stream file is limited to 2 Gigabytes. The stream files are named as [*Stream Base Name*]-[*Stream Serial Number*].pgr. The [*Stream Base Name*] is defined by the user or the application. The [*Stream Serial Number*] is generated internally by the Ladybug library.

For example, if [*Stream Base Name*] is given as 'myStream', Ladybug stream writing API functions will name the stream files as follows:

- myStream-000000.pgr
- myStream-000001.pgr
- myStream-000002.pgr
- Etc. ...

The [*Stream Serial Number*] always begins with 000000. All stream files that have the same [*Stream Base Name*] are considered as subsets of the same Ladybug stream.

When opening a Ladybug stream with a [*Stream Base Name*], the Ladybug API opens all the stream files that have the same [*Stream Base Name*] beginning with 000000.

The total number of images of a Ladybug stream is the sum of all the number of images in each stream file that has the same [*Stream Base Name*].

The data in a stream file is written in the following sequence:

Name	Description
Signature	Ladybug Stream file signature
Stream Header Structure	Information about the stream file
Calibration Data	Camera calibration file
Image 0	First Ladybug image
Image 1	Second Ladybug image
...	...
Image N-1	Last Ladybug image
GPS Summary Data	GPS summary data for the images in this stream file

For more information see Stream File Format.

6.4 Stream File Format

6.4.1 File Signature

Every Ladybug stream file starts with a signature. This signature uniquely identifies the file as a Ladybug stream file.

Offset	Name	Bytes	Type	Value	Description
0x0000	Signature	16	Character string	PGRLADYBUGSTREAM	Ladybug Stream file identifier

6.4.2 Stream Header Structure

The stream header structure begins immediately after the file header at offset 16 from the beginning of the file. It contains the information defined by `LadybugStreamHeadInfo` in `ladybug.h`. The byte order of this data block is little endian.

Offset	Name	Bytes	Type	Description
0x0000	Ladybug stream version no.	4	unsigned int	Stream version number
0x0004	Frame rate	4	unsigned int	The frames recorded per second
0x0008	Base serial No.	4	unsigned int	Ladybug base unit serial number

Offset	Name	Bytes	Type	Description
0x000C	Head serial No.	4	unsigned int	Ladybug head unit serial number
0x0010	Reserved	104	unsigned int	Reserved space
0x0078	Data format	4	unsigned int	Image data format defined in ladybug.h
0x007C	Resolution	4	unsigned int	Image resolution defined in ladybug.h
0x0080	Stippled format	4	unsigned int	Image Bayer pattern
0x0084	Configuration data size	4	unsigned int	Number of bytes of the configuration data
0x0088	N - Number of images	4	unsigned int	Number of images in this stream file
0x008C	M- Number of index	4	unsigned int	Number of entries used in the index table
0x0090	K - Increment	4	unsigned int	Interval value for Indexing the images
0x0094	Stream data offset	4	unsigned int	Offset of the first image data
0x0098	GPS summary data offset	4	unsigned int	Offset of GPS summary data block
0x009C	GPS summary data size	4	unsigned int	Size of GPS summary data block
0x00A0	Frame header size	4	unsigned int	Size of internal frame header.
0x00A4	Humidity availability	4	bool	Whether humidity sensor is available
0x00A8	Humidity minimum	4	unsigned int	Minimum value for sensor
0x00AC	Humidity maximum	4	unsigned int	Maximum value for sensor
0x00B0	Air pressure availability	4	bool	Whether air pressure sensor is available
0x00B4	Air pressure minimum	4	unsigned int	Minimum value for sensor
0x00B8	Air pressure maximum	4	unsigned int	Maximum value for sensor
0x00BC	Compass availability	4	bool	Whether compass sensor is available
0x00C0	Compass minimum	4	unsigned int	Minimum value for sensor
0x00C4	Compass maximum	4	unsigned int	Maximum value for sensor
0x00C8	Accelerometer availability	4	bool	Whether accelerometer sensor is available

Offset	Name	Bytes	Type	Description
0x00CC	Accelerometer minimum	4	unsigned int	Minimum value for sensor
0x00D0	Accelerometer maximum	4	unsigned int	Maximum value for sensor
0x00D4	Gyroscope availability	4	bool	Whether gyroscope sensor is available
0x00D8	Gyroscope minimum	4	unsigned int	Minimum value for sensor
0x00DC	Gyroscope maximum	4	unsigned int	Maximum value for sensor
0x00E0	Frame rate	4	float	Actual frame rate, represented as a floating point value.
0x00E4	Reserved space	780	unsigned int	Reserved space
…	…	…	…	
Image index [M-1]	4	unsigned int	Offset of image (M-1)*K	
…	…	…	…	
0x0BE4	Image index [2]	4	unsigned int	Offset of image 2*K
0x0BE8	Image index [1]	4	unsigned int	Offset of image K
0x0BEC	Image index [0]	4	unsigned int	Offset of image 0.

The image index table between `0x03F0` and `0x0BF0` is used to locate the keyframes of the stream. Using this table can speed up image searching. The value of K (offset `0x0090`) means that the index table contains the offset values for every K^{th} image. The offset values are relative to the beginning of the stream file. For example, if K = 50, the value of 'Image index [5]' is the offset of image 250 (K * 5 = 250). It is the location of image 250 relative to the first byte of the stream file.

6.4.3 Configuration Data

The configuration data begins immediately after stream header structure. The data is in ASCII text format. It is extracted from the Ladybug camera head for image calibration. The size of this data block is the value of 'Configuration Data Size' as defined in the Stream Header Structure.

6.4.4 Frame Header

Since v7 of the frame header, there is a frame header at the start of each image. The size of the frame header can be found in the stream header. Frame headers are present regardless of whether the image data format is JPEG or uncompressed. The information in the frame header can be found in the `LadybugImageHeader` structure.

6.4.5 JPEG Compressed Image Data Structure

If the image format specified for recording is JPEG, each image for the six camera sensors is JPEG compressed in four separate Bayer channels. Therefore, a frame of ladybug image has 24 JPEG data blocks.

The first frame of JPEG images begins immediately after the configuration data. The second frame follows the first frame, the third frame follows the second, and so on. The offset value of the first JPEG image, relative to the beginning of the file, is the value of `Stream Data Offset` as defined in the Stream Header Structure.

The general layout of a JPEG compressed LadybugImage is as follows:

Image Header (0x000 – 0x400)
Cam 0 Bayer 0
Cam 0 Bayer 1
Cam 0 Bayer 2
Cam 0 Bayer 3
Cam 1 Bayer 0
...
Cam 4 Bayer 3
Cam 5 Bayer 0
Cam 5 Bayer 1
Cam 5 Bayer 2
Cam 5 Bayer 3
GPS NMEA data

For each compressed Ladybug image, the GPS NMEA sentences are written at the end of each JPEG image data and are located at the offset value of `GPS_Offset`. If there is no GPS data, `GPS_Offset` and `GPS_Size` are set to zero.

The byte order of this data block is big endian.

Offset	Name	Value	Bytes	Type	Description
0x0000	Timestamp		4	unsigned int	The cycle time seconds, cycle time count and cycle offset of this image
0x0004	Reserved		4	N/A	N/A
0x0008	Data size		4	unsigned int	The total data size of the this frame, including the padding block
0x000C	Reserved		4	N/A	Filled with 0s
0x0010	Fingerprint	0xCAFEBABE	4	Character	Unique fingerprint
0x0014	Version Number		4	unsigned int	Version number
0x0018	Time (seconds)		4	unsigned int	Timestamp, in seconds (UNIX time epoch)

Offset	Name	Value	Bytes	Type	Description
0x001C	Time (microseconds)		4	unsigned int	Microsecond fraction of above second
0x0020	Sequence ID		4	unsigned int	Image sequence number
0x0024	Refresh Rate		4	unsigned int	Horizontal refresh rate
0x0028	Gain[6]		24	unsigned int	Gain values for each camera
0x0040	White balance		4	unsigned int	White balance
0x0044	Bayer gain		4	unsigned int	Same as register 0x1044
0x0048	Bayer map		4	unsigned int	Same as register 0x1040
0x004C	Brightness		4	unsigned int	Brightness
0x0050	Gamma		4	unsigned int	Gamma
0x0054	Head Serial Number		4	unsigned int	Serial number of Ladybug Head
0x0058	Shutter[6]		24	unsigned int	Shutter values for each camera
0x0070	PPS Data		4	unsigned int	Refer to GPS_PPS_TIME_SYNC_ VERIFICATION register on the camera
0x0074	Free space		24	N/A	Reserved space
0x0088	Free space		632	N/A	Random data
0x0300	Free space		56	N/A	Filled with 0s
0x0338	GPS data offset	GPS_Offset	4	unsigned int	The offset of GPS data
0x033C	GPS data size	GPS_Size	4	unsigned int	The size of GPS data
0x0340	JPEG data offset	Offset_0_0	4	unsigned int	Cam 0, Bayer Channel 0
	JPEG data size	Size_0_0	4	unsigned int	Cam 0, Bayer Channel 0
	JPEG data offset	Offset_0_1	4	unsigned int	Cam 0, Bayer Channel 1
	JPEG data size	Size_0_1	4	unsigned int	Cam 0, Bayer Channel 1
	JPEG data offset	Offset_0_2	4	unsigned int	Cam 0, Bayer Channel 2
	JPEG data size	Size_0_2	4	unsigned int	Cam 0, Bayer Channel 2
	JPEG data offset	Offset_0_3	4	unsigned int	Cam 0, Bayer Channel 3

Offset	Name	Value	Bytes	Type	Description
	JPEG data size	Size_0_3	4	unsigned int	Cam 0, Bayer Channel 3
	JPEG data offset	Offset_1_0	4	unsigned int	Cam 1, Bayer Channel 0
	JPEG data size	Size_1_0	4	unsigned int	Cam 1, Bayer Channel 0
...	...	...	...	...	...
	JPEG data offset	Offset_5_2	4	unsigned int	Cam 5, Bayer Channel 2
	JPEG data size	Size_5_2	4	unsigned int	Cam 5, Bayer Channel 2
	JPEG data offset	Offset_5_3	4	unsigned int	Cam 5, Bayer Channel 3
	JPEG data size	Size_5_3	4	unsigned int	Cam 5, Bayer Channel 3
0x0400	JPEG data		...	Binary	...
Offset_i_j	JPEG data		Size_i_j	Binary	Beginning from offset 0x0400 are the 24 JPEG data blocks for Camera *i*, Bayer Channel *j*, where *i = 0, 1, 2, 3, 4, 5, 6* and *j = 0, 1, 2, 3*.
...	...		...	...	...
GPS_Offset	GPS NMEA data		GPS_Size	ASCII text	GPS NMEA sentences

The four bytes of timestamp data at offset 0x0000 are the cycle time seconds, cycle time count and cycle offset when the image is captured.

Description	Cycle Time (seconds)	Cycle Time (ms)	Cycle Offset
Range	0-127	0-7999	0-3071
Bits	0-6	7-19	20-31

For more information about Ladybug time stamp, see the definition of `LadybugTimestamp` struct in ladybug.h.

The data between offset `0x0010` and `0x008F` contains the information of the `LadybugImageInfo` structure defined in `ladybug.h`.

6.4.6 Uncompressed Image Data Structure

If the image format is uncompressed, the image data is the raw binary data from the camera. The first frame begins immediately after the configuration data. The number of bytes for each of the six images is determined by image resolution and data format as defined in the Stream Header Structure.

The Bayer pattern of the image is defined by `Stippled Format` as defined in the Stream Header Structure.

For uncompressed Ladybug images, the GPS NMEA sentences are written to the last 1024 bytes of the image data of camera 5. This means that the last 1024 bytes of image data will be overwritten by GPS data if the GPS device is available.

The following table lists the data structure of each uncompressed image, assuming a resolution of LADYBUG_RESOLUTION_1632x1232.

Offset	Name	Type	Description
0x00000000	Image Data	Binary	Cam-0, Bayer pattern image data
0x001EAE00	Image Data	Binary	Cam-1, Bayer pattern image data
0x003D5C00	Image Data	Binary	Cam-2, Bayer pattern image data
0x005C0A00	Image Data	Binary	Cam-3, Bayer pattern image data
0x007AB800	Image Data	Binary	Cam-4, Bayer pattern image data
0x00996600	Image Data	Binary	Cam-5, Bayer pattern image data
0x00B81400	GPS NMEA data	ASCII text	1024 bytes space for GPS NMEA sentences

6.4.7 GPS Summary Data Format

The GPS summary data begins immediately after the image data discussed in JPEG Compressed or Uncompressed Image. The offset value relative to the beginning of the file is the value of `GPS Summary Data Offset` as defined in the Stream Header Structure. The data structure of the GPS summary data is defined by GPS3DPoint in ladybugstream.h. No other groups are defined in version 1.2 Beta 19 or earlier. The byte order of this data block is big endian.

Offset	Name	Value	Bytes	Type	Description
0x0000	Data identifier	GPSSUMMARY_00001	16	Characters	First group identifier
0x0010	Reserved	Filled with 0's	16	N/A	Reserved space
0x0020	Item data size		4	unsigned int	Size of each data item
0x0024	Number of Items		4	unsigned int	The number items
0x0028	Image No.		4	unsigned int	Associated image number
0x002C	Longitude		8	double	Longitude of item 0
0x0034	Latitude		8	double	Latitude of item 0
0x003C	Altitude		8	double	Altitude of item 0
0x0044	Image No.		4	unsigned int	Associated image number
0x0048	Longitude		8	double	Longitude of item 1
0x0050	Latitude		8	double	Latitude of item 1
0x0058	Altitude		8	double	Altitude of item 1
	…	…	…	…	…
	Image No.		4	unsigned int	Associated image number
	Longitude		8	double	Longitude of item N-1
	Latitude		8	double	Latitude of item N-1
	Altitude		8	double	Altitude of item N-1

6.5 How-To: Using Stream Files

The Ladybug library allows the user to write images captured from a Ladybug camera to disk. Once written to disk, these images can then be extracted for processing.

For code examples, see the **LadybugStreamCopy**, **LadybugSimpleRecording** and **LadybugProcessStream** examples. Examples can be accessed from the Windows **Start** Menu -> **Teledyen Ladybug SDK** -> **Examples**.

For more information about stream files, see Stream File Format.

> **Note:** The most important requirement for working with stream files is that there must be at least 2 GB of free space on the hard drive before writing to disk.

Writing a stream file to disk

The following steps provide a brief overview of how to use the stream functionality of the Ladybug library to write a stream to disk:

- Create a stream context (`LadybugStreamContext`) by calling `ladybugCreateStreamContext()`.
- Initialize the stream context for writing by calling `ladybugInitializeStreamForWriting()`.
- To write an image to disk, simply grab an image, and pass it to `ladybugWriteImageToStream()`.
- When all the writing is complete, call `ladybugStopStream()` to stop writing to disk.
- Destroy the context by calling `ladybugDestroyStreamContext()` when suitable (such as program termination).

Reading a stream file from disk

The following steps provide a brief overview of how to use the stream functionality of the Ladybug library to read a stream from disk:

- Create a stream context (`LadybugStreamContext`) by calling `ladybugCreateStreamContext()`.
- Initialize the stream context for reading by calling `ladybugInitializeStreamForReading()`.
- At this point, additional information about the stream can be obtained by calling `ladybugGetStreamHeader` and `ladybugGetStreamNumOfImages`.
- If a specific image is required, calling `ladybugGoToImage()` will move the stream to the specified image. Otherwise, `ladybugReadImageFromStream()` will retrieve the image from the current reading pointer.
- When reading is complete, call `ladybugStopStream()` to stop reading.
- Destroy the context by calling `ladybugDestroyStreamContext()` when suitable (such as program termination).

Generating Google Maps & Google Earth data

The Ladybug library allows the user to retrieve GPS data from a stream file and automatically generate Google Maps or Google Earth data, which can then be loaded in their respective applications.

If a stream context has already been initialized for reading, calling `ladybugWriteGPSSummaryDataToFile` with the relevant `LadybugGPSFileType` will generate GPS data for the entire stream file.

7 Calibration and Coordinate System

Effective warping and stitching of the images produced by the camera system's six sensors is achieved through accurate calibration of the physical location and orientation of the sensors and the distortion model of the lens. This section discusses the representation used to describe the physical orientation of all of the sensors with respect to one another.

The Ladybug software manages the camera coordinate system by breaking it down into seven right-handed coordinate frames of one of two types: six independent image sensor coordinate frames and a camera coordinate frame.

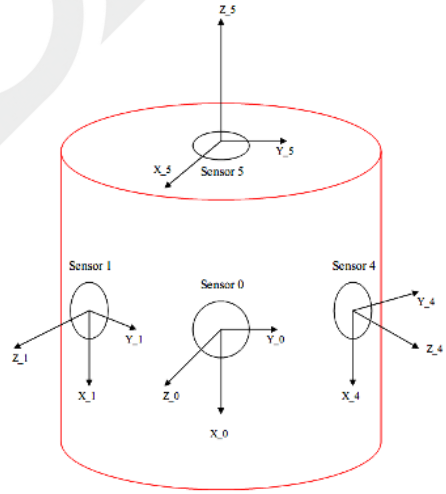
7.1 Coordinate Systems on Ladybug Cameras

Each lens has its own right-handed 3D coordinate system. As well there is a Ladybug 3D Coordinate system that is associated with the camera as a whole. This makes a total of seven 3D coordinate systems on every Ladybug camera. As well, there is a 2D pixel-grid coordinate system for each sensor.

7.1.1 Lens 3D coordinate system

Each of the six lenses has its own 3D coordinate system.

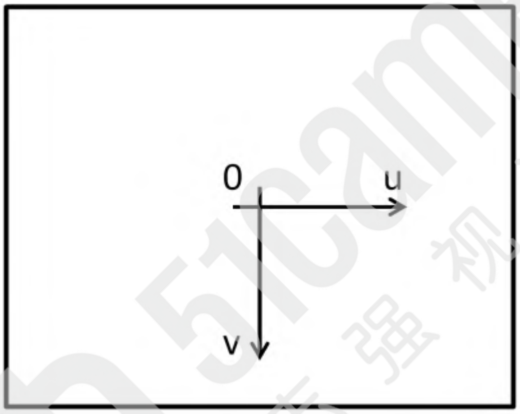
- Origin is the optical center of the lens
- Z-axis points out of the sensor towards the scene – i.e. it is the optical axis
- The X- and Y-axes are relative to the pixel grid of the image sensor associated with that lens
  - The Y-axis points along the image columns. The positive Y direction is in the direction of ascending row number. This points down from the point of view of a normally oriented image
  - The X-axis points along the image rows. The positive X direction is in the direction of ascending column number. This points to the right in a normally oriented image
- This coordinate system is used to represent 3D space from the point-of-view of each lens/sensor pair. Its units are meters, not pixels.



7.1.2 Sensor 2D coordinate system

Each sensor has its own 2D coordinate system.

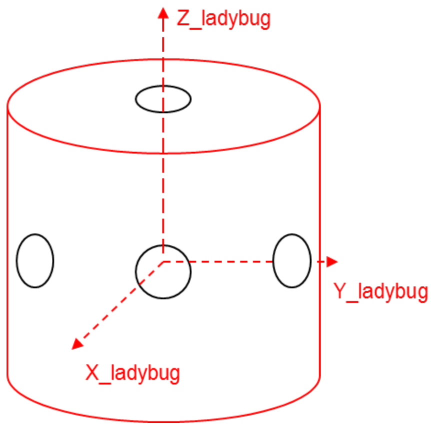
- The u- and v-axes are the image based 2D image coordinate system for the rectified image space and are measured in pixels
- The origin of the coordinate system is at the intersection of the optical axis and the rectified image plane and differs for each sensor
- The u-axis points along the rows of the image sensor in the direction of ascending column number (i.e. to the right)
- The v-axis points along the columns in the direction of ascending row number (i.e. down).



7.1.3 Ladybug Camera Coordinate System

The Ladybug Camera coordinate system is centered within the Ladybug case and is determined by the position of the 6 lens coordinate systems.

- Origin is the center of the five horizontal camera origins
- Z-axis is parallel to the optical axis of the top lens (lens 5) (*)
- X-axis is parallel to the optical axis of lens 0 (*)
- Y-axis is consistent with a right handed coordinate system based on the X- and Z-axes
- There may be some variations between Ladybug models
- (*) Note – due to assembly tolerances the optical axes of lens 5 and lens 0 will typically not be perfectly perpendicular. The X-axis of the Ladybug Camera coordinate system is adjusted slightly to ensure that they are perpendicular.



For more detailed information on the representation used to describe the physical orientation of all of the sensors with respect to one another and provides instructions for transforming 2D local points to 3D global points and vice versa, see Geometric Vision using Ladybug Cameras.

7.2 Projection Types

Once a three-dimensional spherical coordinate system is obtained, the image on the sphere can be projected to a mapping based on different projection methods. The projected image is usually two-dimensional so that it can easily be displayed on a monitor or printed on paper. Each projection type has its own benefits and shortcomings.

Radial (Equirectangular) Projection

This is one of the most popular projections, and the output image is easy to use. In LadybugCapPro, you can output video to this projection by using the Stream Processing Toolbar and selecting Output Type "Panoramic." Then, using the Image Processing Toolbar, specify Mapping Type "Radial."

The projected image has two coordinates – theta for horizontal, and phi for vertical. The projection equation from the spherical coordinate system is as follows:

$$R = \sqrt{X^2 + Y^2 + Z^2}$$

$$\theta = ATAN2(Y, X)$$

$$\varphi = ACOS(\frac{Z}{R})$$

Where (X, Y, Z) are points on the spherical coordinate system, and ATAN2 and ACOS are functions provided by the standard C library.

(Θ,Φ) are coordinates of the projected image.

The range of values of Θ is –Pi to Pi.

The range of values of Φ is 0 to Pi.

In order to convert to the actual pixel position on the radial projection image, appropriate scaling is needed based on these value ranges.

(ϴ,Φ) can be obtained by referring to the fTheta and fPhi members of the LadybugPoint3d struct, which is obtained by invoking ladybugGet3dMap().

The benefit of this projection is that all the points in the original spherical coordinate system can be mapped on a single image. Additionally, the correspondence of the original point and the projected point is simple, in that the horizontal axis corresponds to longitude and the vertical axis corresponds to latitude of a globe. However, this projection suffers from the disadvantage of pixels becoming increasingly stretched out as one approaches the poles of the sphere (top and bottom of the projected image).

Example radial projection image

Cylindrical Projection

This projection is similar to the radial projection, but with a limited field of view, as areas close to the poles are not able to be rendered. In LadybugCapPro, you can output video to this projection by using the Stream Processing Toolbar and selecting Output Type "Panoramic." Then, using the Image Processing Toolbar, specify Mapping Type "Cylindrical."

The projection equation is as follows:

$$R = \sqrt{X^2 + Y^2 + Z^2}$$

$$\theta = ATAN2(Y, X)$$

$$\varphi = -\frac{ZR}{\sqrt{R^2 - Z^2}}$$

(Θ,Φ) are the coordinates of the projected image.

Θ is computed in the same manner as in the radial projection.

Φ can go to infinity as the 3D point nears the pole, which is why the field of view of the cylindrical projection must be limited. When rendered using LadybugCapPro, only the field of view between -45 degrees and 45 degrees is displayed. Thus this projection is useful when only the images from the side cameras are needed.

(Θ,Φ) can be obtained by referring to the fCylAngle and fCylHeight members of the LadybugPoint3d struct, which is obtained by invoking ladybugGet3dMap().

Example cylindrical projection image

Dome Projection

This is a projection that maps the sphere to a dome-like shape. In LadybugCapPro, select Output Type "Dome." The projection equation is as follows:

$$R = \sqrt{X^2 + Y^2 + Z^2}$$

$$\theta = ATAN2\,(Y, X)$$

$$\varphi = -\frac{Z \times R}{\sqrt{R^2 - Z^2}}$$

$$U = -\frac{\varphi}{\varphi Max} \times \sin\theta$$

$$V = -\frac{\varphi}{\varphi Max} \times \cos\theta$$

Where U ranges between -1 and 1, V ranges between -1 and 1 and √(U^2+V^2) ≤1.

U and V are zero at the center of the image with U increasing to the right and V increasing upwards.

The amount of the image drawn can be controlled by ϕMax which can be set and get via `ladybugChangeDomeViewAngle()` and `ladybugGetDomeViewAngle()`.

In the above formulas, ϕMax is measured in radians.

Example dome projection image

Cubic (Skybox) Projection

This projection requires 6 images, where each image is the surface of a cube. By dividing the entire sphere into 6 images, the distortion in each image is limited. To reconstruct the panoramic image, the 6 cube-suface images must be displayed using 3D computer graphics. Thus, this mapping is suitable for video game applications. The projection equation is as follows:

$$U = \frac{X}{Z}$$

$$V = \frac{Y}{Z}$$

This projection is equivalent to the view captured by a lens that has no distortion (pin-hole lens). Each surface of the cube can be obtained by rendering the spherical image while setting the appropriate camera rotation and field of view to exactly 90 degrees. This is achievable by calling ladybugSetSphericalViewParams(). To use LadybugCapPro, select Output Type "6 Cube map."

Example cubic projection images

8 Ladybug API Overview

The Ladybug API has a help file derived from the Ladybug header (.h) files.

Help file—C:\Program Files\Teledyne\Ladybug\doc

Header files—C:\Program Files\Teledyne\Ladybug\include

- ladybug.h
- ladybuggeom.h
- ladybuggps.h
- ladybugrenderer.h
- ladybugstream.h
- pgrcameragui.h

8.1 API functionality

The Ladybug API can be used to:

- Stream images off the camera for the purposes of viewing.
- Stream images off the camera to disk to be viewed later.
- Control the various camera parameters including shutter, gain and white balance.
- Color process images using a variety of different color processing algorithms.
- Control the level of JPEG compression.
- Control the recording of images.
- Access and download images in stream files to the host computer.
- Display fully stitched panoramic and spherical images in real time or from a saved file.
- Interface with a GPS device and embed the resulting GPS data into images.

A Ladybug API Examples

The following examples are included in the Ladybug SDK.

Examples are accessible from:

Start Menu -->Teledyne Ladybug SDK--> Examples

With the exception of ladybugCSharpEx and ladybugProcessStream_CSharp, all examples are Visual C++.

Example	Description
ladybug3dViewer	Shows how to display a spherical view in which the user can pan and tilt the image inside the sphere.
ladybugAdvancedRenderEx	Shows how to draw a Ladybug 3D spherical image in conjunction with other 3D objects.
ladybugCaptureHDRImage	Demonstrates how to capture a series of images closely spaced in time suitable for input into a high dynamic range image creation system.
ladybugCSharpEx	Shows how to create a C# program that uses the Ladybug API.
ladybugEnvironmentalSensors	Shows how to access the information from the environmental sensors.
ladybugEnvMap	Shows how to apply cube mapping on spherical images to construct a skybox.
ladybugGPSTimeSync	Shows how to use the GPS Time Synchronization functionality.
ladybugOGLTextureEx	Shows how to access Ladybug images directly on the graphics card as an OpenGL texture map.
ladybugOutput3DMesh	Demonstrates how to produce a 3D mesh out of calibration data from the connected camera.
ladybugPanoramic	Shows how to use a document-view application to grab Ladybug images and display them in a window.
ladybugPanoStitchExample	Shows how to extract an image set from a Ladybug camera, stitch it together and write the final stitched image to disk.
ladybugPostProcessing	Shows how to perform post processing on 12- or 16-bit images.
ladybugProcessStream	Shows how to process all, or part, of a stream file. Also available as a C# example: ladybugProcessStream_CSharp.
ladybugProcessStreamParallel	Shows how to process a Ladybug image stream using multiple Ladybug context parallel processing.
ladybugSimpleGPS	Shows how to use a GPS device in conjunction with a Ladybug camera to integrate GPS data with Ladybug images.
ladybugSimpleGrab	Illustrates the basics of acquiring an image from a Ladybug camera.
ladybugSimpleGrabDisplay	Shows how to use the OpenGL Utility Toolkit (GLUT) to grab Ladybug images and display them in a simple window.

Example	Description
ladybugSimpleRecording	Shows how to record Ladybug images to .pgr stream files. When used in conjunction with a GPS device, also shows how to record images when the GPS location changes after a specified distance.
ladybugStitchFrom3DMesh	Shows how to stitch six raw images without using the Ladybug SDK.
ladybugStreamCopy	Copies images from a Ladybug source stream to a destination stream.
ladybugTranslate2dTo3d	Shows how to translate a 2D point in the raw image to a 3D point.
ladybugTriggerEx	Shows how to control trigger and strobe.

A.1 ladybug3dViewer

This example shows how to display a spherical view in which the user can pan and tilt the image inside the sphere. The program reads one rectangular panoramic image (.bmp or .ppm) and maps it onto the sphere using OpenGL functions.

This program does not handle video or .pgr format stream files, and does not require the Ladybug SDK API.

A.2 ladybugAdvancedRenderEx

This example shows how to draw a Ladybug 3D spherical image in conjunction with other 3D objects. To render 3D objects together with a Ladybug 3D spherical image, ladybugDisplayImage() must be called prior to drawing any objects. The size and position of the objects must be inside the Ladybug spherical image. Otherwise, the objects will not be seen. Additionally, the OpenGL depth test must be enabled.

Note:
This example must be run with glut32.dll.
This example must open the following .ppm texture files:

- TextureCam0.ppm
- TextureCam1.ppm
- TextureCam2.ppm
- TextureCam3.ppm
- TextureCam4.ppm
- TextureCam5.ppm

A.3 ladybugCaptureHDRImage

This example code demonstrates how to capture a series of images closely spaced in time suitable for input into a high dynamic range image creation system.

The main() function initializes the camera and calls the other subroutines.

The setupHDRRegisters() subroutine sets all of the registers necessary to put the camera into 'HDR Mode'.

captureImages() captures images directly from a Ladybug camera.

processImages() computes the panoramic images.

The Ladybug has a bank of four gain and shutter registers in addition to its standard set. When put into 'HDR Mode', the camera cycles through the settings contained in these registers on an image by image basis. This allows users to capture a set of four images with widely varying exposure settings. The four images can be captured within 4/30 of a second if the data format is set to LADYBUG_DATAFORMAT_COLOR_SEP_SEQUENTIAL_JPEG.

The shutter and gain values are read from an INI file defined by INI_FILE_NAME. If you find the shutter and gain settings are not appropriate, change the data in this file.

Once these images have been captured, the program processes the images and outputs a configuration file containing exposure data suitable for input into a program such as 'pfstools' and 'pfscalibration'.

Having captured the images, the user should then run the 'pfsinhdrgen' program in the image directory with a command line similar to the following:

```
pfsinhdrgen HDRDescription.hdrgen | pfshdrcalibrate -v | pfsout output.hdr
```

Where 'HDRDescription.hdrgen' is the name of the configuration file output by this program and 'output.hdr' is the name of the output image.

Then the output file can be viewed using pfsview (or pfsv).

```
pfsv output.hdr
```

You can also make an HDR image out of four output images using Adobe Photoshop CS3, easyHDR, etc. In this case, you don't need to provide additional exposure data.

'pfstools' is available at http://pfstools.sourceforge.net/

'pfscalibration' is available at: http://www.mpi-inf.mpg.de/resources/hdr/calibration/pfs.html

A.4 ladybugCSharpEx

This example shows how to create a C# program that uses the Ladybug API. The program can display stitched panoramic images either from the camera or from a stream file. If a stream file contains GPS information, it is also displayed. The program requires LadybugAPI.cs and LadybugAPI_GPS.cs, which define the interface of the Ladybug API for the C# language.

A.5 ladybugEnvironmentalSensors

This example shows how to use the Ladybug API to obtain data from the environmental sensors on the Ladybug5. In addition to reading the raw information from the camera, the example shows how to calculate the camera's heading from the raw compass values.

A.6 ladybugEnvMap

This example illustrates how to apply cube mapping on Ladybug's spherical images to construct a skybox. In computer graphics, cube mapping is a type of environment mapping used to simulate surfaces that reflect the scene at a distant location.

Here, Ladybug images are used as the environment and are updated in real time. For each scene, six surfaces of a cube are rendered. This is done by rendering Ladybug's spherical view 6 times, setting the field of view to 90 degrees and positioning the virtual camera to specific surface directions. These rendering results are then used as textures for the cube mapping. The overall scene, which is comprised of the reflective objects, is then rendered. All calculations required to construct the cube map are handled inside the OpenGL library.

This example must be run with freeglut.dll present.

Note: This example must be run with freeglut32.dll present.

A.7 ladybugGPSTimeSync

This example shows how to use the GPS Time Synchronization functionality. This includes enabling and disabling the GPS Time Sync, setting an appropriate baud rate and checking the status of the time synchronization. See Using PPS to Synchronize with External GPS for more information on synchronizing Ladybug with an external GPS device.

A.8 ladybugOGLTextureEx

This example shows how to access Ladybug images directly on the graphics card as an OpenGL texture map. The rendered Ladybug image is accessed by its texture ID and can be mapped to any geometric object as desired by using OpenGL functions.

Right click the mouse in the client area to display a menu and select various Ladybug image types.

Note: This example must be run with glut32.dll.

A.9 ladybugOutput3DMesh

This example demonstrates how to produce a 3D mesh out of calibration data from the connected camera. The output of this program can be directly used for the input of the program ladybugStitchFrom3DMesh. You can save the output of this program to a file by using redirection. From the command prompt, navigate (cd) to the Ladybug SDK's "bin" directory and type ladybugoutput3dmesh >mymesh.txt. You will then have the output in the file "mymesh.txt".

A.10 ladybugPanoramic

This example shows how to use a document-view application to grab Ladybug images and display them in a window.

The CLadybugPanoramicDoc class is used to initialize and start a Ladybug camera. It creates a thread for grabbing and processing images.

The CLadybugPanoramicView class is used to display Ladybug images. It initializes the window for OpenGL display. To display a ladybug image, CLadybugPanoramicView::OnDraw() calls the image-drawing API functions.

A.11 ladybugPanoStitchExample

This example shows how to extract an image set from a Ladybug camera, stitch it together and write the final stitched image to disk.

Since Ladybug library version 1.3.alpha.01, this example is modified to use ladybugRenderOffScreenImage(), which is hardware accelerated, to render the stitched images.

Typing ladybugPanoStitchExample /? (or ? -?) at the command prompt prints the usage information of this application.

A.12 ladybugPostProcessing

This example shows how to use the post processing pipeline together with a 12- or 16-bit image to produce a processed image. The code shows how to modify the LadybugAdjustmentParameters structure to define the type of post processing to perform.

A.13 ladybugProcessStream

This example shows how to process all, or part, of a stream file. The program processes each frame and outputs an image file sequentially. If the stream file contains GPS information, the program outputs the information to a separate text file. By editing this source code, users can change image size, image type, output file format, color processing algorithm, and blending width. Users can also change options for falloff correction, software rendering and stabilization.

> **Note:** This example is also available in C# as ladybugProcessStream_CSharp.

A.14 ladybugProcessStreamParallel

This example shows how to process a Ladybug image stream using multiple Ladybug context parallel processing.

This program creates a stream reading thread and one or more image processing threads. The stream reading thread reads images from a stream and puts them into a buffer queue. Each processing thread gets images from the buffer queue and processes the images concurrently with other threads.

The number of processing threads you can create depends on many factors such as image resolution, color processing method, the size of the rendered image and the size of the graphics card memory.

If the required resources are beyond the ability of the graphics card, the program may report a run-time error.

The overall processing speed depends on several factors: disk I/O speed, number of CPUs and performance of the graphics card. For fast stream processing, we recommend the following:

- Multi-core processor machine
- Graphics card with 512 Mbytes memory or more
- Fast hard disk drive configuration, such as RAID0
- Reading the stream from one drive and writing the rendered images to another drive.

This example reads the processing parameter options from the command line. Use -? or -h to display the usage help.

A.15 ladybugSimpleGPS

This example shows how to use a GPS device in conjunction with a Ladybug camera to integrate GPS data with Ladybug images.

Before running this example, you need to know the COM port to which the GPS device is mapped, even if the device uses a USB interface. Right click on "My Computer" from the Windows Start menu. Under the "Hardware" tab, click "Device Manager." Expand the "Ports (COM & LPT)" node and note the COM port to which the GPS device is mapped.

A.16 ladybugSimpleGrab

This example illustrates the basics of acquiring an image from a Ladybug camera. The program performs the following tasks:

1. Creates a context.
2. Initializes a camera.
3. Starts the transmission of images.
4. Grabs an image.
5. Processes the grabbed image using a color processing algorithm.
6. Saves the 6 raw images as BMP files.
7. Destroys the context.

A.17 ladybugSimpleGrabDisplay

This example shows how to use OpenGL Utility Toolkit (GLUT) to grab Ladybug images and display them in a simple window. This example starts the first Labybug camera on the bus. The camera is started in JPEG mode and the images are processed with the LADYBUG_DOWNSAMPLE4 color processing method.

Right click the mouse in the client area to display a menu and select various Ladybug image types.

> **Note:** This example must be run with glut32.dll.

A.18 ladybugSimpleRecording

This example shows how to record Ladybug images to .pgr stream files. The example starts the first Ladybug camera on the bus with the parameters in the .ini file defined by INI_FILENAME.

This example displays the grabbed images only when the grabbing function returns LADYBUG_TIMEOUT. This means that saving images is the highest priority.

Right click the mouse in the client area to popup a menu and select various options, or use the following hot keys:

- 'r' or 'R' - start recording, press again to stop recording.
- 'p' or 'P' - display panoramic image.
- 'a' or 'A' - display all-camera image.
- 'd' or 'D' - display dome view image.
- 'Esc', 'q' or 'Q' - exit the program.

When used in conjunction with a GPS device, this example also shows how to record images when the GPS location changes after a specified distance, in meters. The distance parameter is specified in the .ini file. The accuracy of the result depends on the GPS device and the GPS data update rate.

> **Note:** This example must be run with freeglut.dll and Ladybug SDK v. 1.3.0.2 or later.

A.19 ladybugStitchFrom3DMesh

This example shows how to stitch six raw images without using the Ladybug SDK. Note that users still need the 3D mesh data produced by the program ladybugOutput3DMesh, which requires the Ladybug SDK.

This program is useful for users who want to stitch images in an environment where the Ladybug SDK is not supported.

A.20 ladybugStreamCopy

This program copies images from a Ladybug source stream to a destination stream. If a calibration file is specified, this program writes this calibration file to the destination file instead of using the calibration file in the source stream.

The last two arguments specify how many images to copy. If they are not specified, all the images are copied.

A.21 ladybugTranslate2dTo3d

This example shows how to use the Ladybug API to translate a 2D point in the raw image to a 3D point in the Ladybug camera coordinate space and vice versa. It also shows how to use ladybugGet3dMap() provided by the Ladybug API to perform the translation.

This example is a companion to Geometric Vision using Ladybug Cameras.

A.22 ladybugTriggerEx

This example shows how to use the Ladybug API to control the trigger and strobe functionality of the camera. The example sets the camera into trigger mode 0 (Standard) and then uses software triggering to trigger when an image is captured.

B Control and Status Registers

Some features of the Ladybug are accessible only using control and status registers (CSRs) that conform to the IICD 1.32 standard.

These include the following:

- Bayer tile mapping—BAYER_TILE_MAPPING: 1040h
- Flash data—DATA_FLASH_CTRL: 1240h
- Frame buffer—IMAGE_RETRANSMIT: 634h
- Gain—GAIN: 820h
- Imaging parameters—Imaging Parameters: 800h-888h
- Independent control—INDEPENDENT_CONTROL_INQ: 1E94h
- JPEG buffer—JPEG_BUFFER_USAGE: 1E84h
- JPEG compression—JPEG_CTRL: 1E80h
- JPEG quality—JPEG_MAX_QUALITY: 1E8Ch
- Memory channels—Memory Channel Registers
- Pixel defect correction—PIXEL_DEFECT_CTRL: 1A60h
- Shutter—SHUTTER: 81Ch
- Software trigger—SOFTWARE_TRIGGER: 62Ch
- Trigger delay—TRIGGER_DELAY: 834h
- Trigger modes—TRIGGER_MODE: 830h

A complete list of CSRs can be found in the *FLIR Machine Vision Camera Register Reference*.

B.1 AE_ROI: 1A70 – 1A74h

AE_ROI is not supported on quad tap sensor cameras.

> **Note:**
> To calculate the base address for an offset CSR:

> 1. Query the offset inquiry register.
> 2. Multiple the value by 4. (The value is a 32-bit offset.)
> 3. Remove the 0xF prefix from the result. (i.e., F70000h becomes 70000h)

Format:

Offset	Name	Field	Bit	Description
1A70h	AE_ROI_CTRL	Presence_Inq	[0]	Presence of this feature 0:Not Available, 1: Available
			[1-5]	Reserved
		ON_OFF	[6]	Read: read a status Write: ON or OFF for this feature 0: OFF, 1: ON If this bit = 0, other fields will be read only
			[7-31]	Reserved
1A74h	AE_ROI_OFFSET		[0-31]	32-bit offset for the AE_ROI CSRs
Base + 0h	AE_ROI_UNIT_POSITION_INQ	Hposunit	[0-15]	Horizontal units for position
		Vposunit	[16-31]	Vertical units for position
Base + 4h	AE_ROI_UNIT_SIZE_INQ	Hunit	[0-15]	Horizontal units for size
		Vunit	[16-31]	Vertical units for size
Base + 8h	AE_ROI_POSITION	Left	[0-15]	Left position of ROI
		Top	[16-31]	Top position of ROI
Base + Ch	AE_ROI_SIZE	Width	[0-15]	Width of ROI
		Height	[16-31]	Height of ROI

B.2 AUTO_EXPOSURE: 804h

> **Note:** Formulas for converting the fixed point (relative) values to floating point (absolute) values are not provided. Users wishing to work with real-world values should refer to Absolute Value CSRs .

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Abs_Control	[1]	Absolute value control 0: Control with the *Value* field, 1: Control with the Absolute value CSR. If this bit = 1, the value in the *Value* field is ignored.
	[2-4]	Reserved
One_Push	[5]	One push auto mode (controlled automatically by camera only once) Read: 0: Not in operation, 1: In operation Write: 1: Begin to work (self-cleared after operation) If A_M_Mode = 1, this bit is ignored
ON_OFF	[6]	Read: read a status Write: ON or OFF for this feature 0: OFF, 1: ON If this bit = 0, other fields will be read only
A_M_Mode	[7]	Read: read a current mode Write: set the mode 0: Manual, 1: Automatic
High_Value	[8-19]	Upper 4 bits of the shutter value available only in extended shutter mode (outside of specification).
Value	[20-31]	Value. A write to this value in 'Auto' mode will be ignored.

B.3 BAYER_TILE_MAPPING: 1040h

This 32-bit read only register specifies the sense of the cameras' Bayer tiling. Various colors are indicated by the ASCII representation of the first letter of their name.

Color	ASCII
Red (R)	52h
Green (G)	47h
Blue (B)	42h
Monochrome (Y)	59h

For example, 0x52474742 is RGGB and 0x59595959 is YYYY.

Note: Because color models support on-board color processing, the camera reports YYYY tiling when operating in any non-raw Bayer data format. For more information, see .

Format

Field	Bit	Description
Bayer_Sense_A	[0-7]	ASCII representation of the first letter of the color of pixel (0,0) in the Bayer tile.
Bayer_Sense_B	[8-15]	ASCII representation of the first letter of the color of pixel (0,1) in the Bayer tile.
Bayer_Sense _C	[16-24]	ASCII representation of the first letter of the color of pixel (1,0) in the Bayer tile.
Bayer_Sense _D	[25-31]	ASCII representation of the first letter of the color of pixel (1,1) in the Bayer tile.

B.4 DATA_FLASH_CTRL: 1240h

This register controls access to the camera's on-board flash memory. Each bit in the data flash is initially set to 1.

The user can transfer as much data as necessary to the offset address (1244h), then perform a single write to the control register to commit the data to flash. Any modified data is committed by writing to this register, or by accessing any other control register.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-5]	Reserved
Clean_Page	[6]	Read: 0: Page is dirty, 1: Page is clean Write: 0: No-op, 1: Write page to data flash
	[7]	Reserved
Page_Size	[8-19]	8 == 256 byte page 9 == 512 byte page
Num_Pages	[20-31]	11 == 2048 pages 13 == 8192 pages

B.5 DATA_FLASH_DATA: 1244h

This register provides the 32-bit offset to the start of where the data is stored in the flash memory.

Format:

Offset	Field	Bit	Description
1244h	DF_Data	[0-31]	32-bit offset to the start of data

B.6 FRAME_INFO: 12F8h

Field	Bit	Description	Frame-Specific Information
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available	
	[1-5]	Reserved	

Field	Bit	Description	Frame-Specific Information
ROI_Pos_Inq	[6]	Presence of image-specific information display 0: Not Available, 1: Available	
GPIO_State_Inq	[7]		
Strobe_Pat_Inq	[8]		
Frame_Count_Inq	[9]		
WB_CSR_Inq	[10]		
Exp_CSR_Inq	[11]		
Bright_CSR_Inq	[12]		
Shutter_CSR_Inq	[13]		
Gain_CSR_Inq	[14]		
Time_Inq	[15]		
CSR_Abs_Value	[16]	Toggles between displaying 32-bit relative or absolute CSR values. If absolute value not supported, relative value is displayed. 0: Relative, 1: Absolute This field is currently read-only	
	[17-21]	Reserved	
Insert_Info	[22]	Display image-specific information 0: Off 1: On	Region of Interest (ROI) position
	[23]		GPIO Pin State
	[24]		Strobe Pattern Counter
	[25]		Frame Counter
	[26]		White Balance CSR
	[27]		Exposure CSR
	[28]		Brightness CSR
	[29]		Shutter Value
	[30]		Gain CSR
	[31]		Timestamp

B.7 GAIN: 820h

The value field in this register can be set in three ways:

Method	Description
Absolute	The user sets the value is set via the absolute register. The *Value* field becomes read only and reflects the converted absolute value.
Manual	The user sets the value in the *Value* field. The absolute register becomes read only and contains the current value.
Automatic	The value is set automatically by another register and both the *Value* field and the absolute register become read only.

Note: Formulas for converting the fixed point (relative) values to floating point (absolute) values are not provided. Users wishing to work with real-world values should refer to Absolute Value CSRs .

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Abs_Control	[1]	Absolute value control 0: Control in the Value field, 1: Control in the Absolute value CSR. If this bit = 1, the value in the Value field is read-only.
	[2-4]	Reserved
One_Push	[5]	One push auto mode (controlled automatically only once) Read: 0: Not in operation, 1: In operation Write: 1: Begin to work (self-cleared after operation) If A_M_Mode = 1, this bit is ignored
ON_OFF	[6]	Read: read a status Write: ON or OFF for this feature 0: OFF, 1: ON If this bit = 0, other fields will be read only
A_M_Mode	[7]	Read: read a current mode Write: set the mode 0: Manual, 1: Automatic
	[8-19]	Reserved
Value	[20-31]	Value. A write to this value in 'Auto' mode will be ignored.

B.8 HDR: 1800h – 1884h

This register allows the user to access and control a multiple exposure quick cycle mode, which is useful for high dynamic range (HDR) imaging.

Note that if bit [31] of the FRAME_INFO: 12F8h is set to 1, the camera will embed the current shutter/gain value in the image when bit [6] of HDR_CTRL is set to 1. The image timestamp will be embedded in the first 32-bits of image data, the shutter value in the second 32-bits, and gain in the third, all in big-endian format.

Note that the on/off bit for the HDR shutter and gain registers is hard-coded to on.

Format:

Offset	Name	Field	Bit	Description
1800h	HDR_CTRL	Presence_Inq	[0]	Presence of this feature 0: Not available, 1: Available
		-	[1-5]	Reserved
		ON_OFF	[6]	Read: read a status Write: ON or OFF for this feature 0: OFF, 1: ON If this bit = 0, other fields will be read only
		-	[7-31]	Reserved
1820h	HDR_SHUTTER_0	Presence_Inq	[0]	Presence of this feature 0: Not available, 1: Available
		-	[1-19]	Reserved
		Value	[20-31]	Query SHUTTER_INQ register 51Ch for range of possible shutter values
1824h	HDR_GAIN_0	Presence_Inq	[0]	Presence of this feature 0: Not available, 1: Available
		-	[1-19]	Reserved
		Value	[20-31]	Query GAIN_INQ register 520h for range of possible gain values
1840h	HDR_SHUTTER_1	Same format as HDR_SHUTTER_0		
1844h	HDR_GAIN_1	Same format as HDR_GAIN_0		
1860h	HDR_SHUTTER_2	Same format as HDR_SHUTTER_0		
1864h	HDR_GAIN_2	Same format as HDR_GAIN_0		
1880h	HDR_SHUTTER_3	Same format as HDR_SHUTTER_0		
1884h	HDR_GAIN_3	Same format as HDR_GAIN_0		

B.9 Imaging Parameters: 800h-888h

The following imaging parameters share the same register format.

Parameter	Register
Brightness	800h
Sharpness	808h
Hue	810h
Saturation	814h
Gamma	818h
Gain	820h
Iris	824h
Focus	828h
Pan	884h
Tilt	888h

These imaging parameters are defined by **modes** and **values**.

There are three modes:

Mode	Description
On/Off	Determines if the feature is on. If off, values are fixed and not controllable.
Auto/Manual	If the feature is on, determines if the feature is in automatic or manual mode. If manual, values can be set.
One Push	If the feature is in manual mode, the camera executes once automatically and then returns to manual mode.

The value field in this register can be set in three ways:

Method	Description
Absolute	The user sets the value is set via the absolute register. The *Value* field becomes read only and reflects the converted absolute value.
Manual	The user sets the value in the *Value* field. The absolute register becomes read only and contains the current value.
Automatic	The value is set automatically by another register and both the *Value* field and the absolute register become read only.

Note: Formulas for converting the fixed point (relative) values to floating point (absolute) values are not provided. Users wishing to work with real-world values should refer to Absolute Value CSRs .

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Abs_Control	[1]	Absolute value control 0: Control in the Value field, 1: Control in the Absolute value CSR. If this bit = 1, the value in the Value field is read-only.
	[2-4]	Reserved
One_Push	[5]	One push auto mode (controlled automatically only once) Read: 0: Not in operation, 1: In operation Write: 1: Begin to work (self-cleared after operation) If A_M_Mode = 1, this bit is ignored

Field	Bit	Description
ON_OFF	[6]	Read: read a status Write: ON or OFF for this feature 0: OFF, 1: ON If this bit = 0, other fields will be read only
A_M_Mode	[7]	Read: read a current mode Write: set the mode 0: Manual, 1: Automatic
	[8-19]	Reserved
Value	[20-31]	Value. A write to this value in 'Auto' mode will be ignored.

B.10 IMAGE_RETRANSMIT: 634h

This register provides an interface to the camera's frame buffer functionality.

Transmitting buffered data is available when continuous shot is disabled. Either One shot or Multi shot can be used to transmit buffered data when *Transfer_Data_Select* = 1. Multi shot is used for transmitting one or more (as specified by *Count_Number*) buffered images. One shot is used for retransmission of the last image from the retransmit buffer.

Image data is stored in a circular image buffer when *Image_Buffer_Ctrl* = 1. If the circular buffer overflows, the oldest image in the buffer is overwritten.

Transmitted data is always stored in the retransmit buffer. If a last or previous image does not exist, (for example, an image has not been acquired since a video format or mode change), the camera still transmits an image from the retransmit buffer, but its contents are undefined.

The image buffer is initialized when *Image_Buffer_Ctr* is written to '1'. Changing the video format, video mode, image_ size, or color_coding causes the image buffer to be initialized and *Max_Num_Images* to be updated.

Format:

Field	Bit	Description
Image_Buffer_Ctrl	[0]	Image Buffer On/Off Control 0: OFF, 1: ON
Transfer_Data_Select	[1]	Transfer data path 0: Live data, 1: Buffered image data Ignored if ISO_EN=1
	[2-7]	Reserved
Max_Num_Images	[8-19]	Maximum number of images that can be stored in the current video format. Must be greater than zero. This field is read only.
Number_of_Images	[20-31]	The number of images currently in buffer. This field is read only.

B.11 INDEPENDENT_CONTROL_INQ: 1E94h

This register gives the 32-bit offset of the base address of the SUB_SHUTTER_*, SUB_GAIN_* and SUB_AUTO_ EXPOSURE_* CSRs.

Format:

Offset	Name	Bit	Description
1E94h	INDEPENDENT_CONTROL_INQ	[0..31]	32-bit offset of the Independent Sensor Control CSRs.

Format:

Offset	Name	Field	Bit	Description
Base + 00h	SUB_GAIN_0	Presence_Inq	[0]	Presence of this feature 0: Not available, 1: Available
			[1-5]	Reserved
		ON_OFF	[6]	Write: ON or OFF for this feature Read: read a status 0: OFF, 1: ON If this bit = 0, other fields are read only. This bit is shared across all SUB_GAIN_* registers; it cannot be set independently. Setting this bit effectively sets the equivalent bit of Gain CSR 820h to the inverse value, and vice versa. Setting this bit effectively sets the equivalent bit of the Sub_Auto_Exposure_* CSRs to the inverse value.
		A_M_Mode	[7]	Write: set the mode Read: read a current mode 0: Manual, 1: Automatic
			[8-19]	Reserved
		Value	[20-31]	Gain value of Camera_0. A write to this field in ‘Auto’ mode is ignored.

Offset	Name	Field	Bit	Description
Base + 04h	SUB_SHUTTER_0	Presence_Inq	[0]	Presence of this feature 0: Not available, 1: Available
			[1-5]	Reserved
		ON_OFF	[6]	Write: ON or OFF for this feature Read: read a status 0: OFF, 1: ON If this bit = 0, other fields are read only. This bit is shared across all SUB_SHUTTER_* registers; it cannot be set independently. Setting this bit effectively sets the equivalent bit of Shutter CSR 81Ch to the inverse value, and vice versa. Setting this bit sets the equivalent bit of the Sub_Auto_Exposure_* CSRs to the inverse value.
		A_M_Mode	[7]	Write: set the mode Read: read a current mode 0: Manual, 1: Automatic
			[8-19]	Reserved
		Value	[20-31]	Shutter value of Camera_0. A write to this field in 'Auto' mode is ignored.
Base + 08h	SUB_AUTO_EXPOSURE_0	Presence_Inq	[0]	Presence of this feature 0: Not available, 1: Available
		-	[1-5]	Reserved
		ON_OFF	[6]	Write: ON or OFF for this feature Read: read a status 0: OFF, 1: ON If this bit = 0, other fields are read only. This bit is shared across all SUB_AUTO_EXPOSURE_* registers; it cannot be set independently. Setting this bit effectively sets the equivalent bits of SUB_SHUTTER_* and SUB_GAIN_* CSRs to the inverse value.
		A_M_Mode	[7]	Write: set the mode Read: read a current mode 0: Manual, 1: Automatic
			[8-19]	Reserved
		Value	[20-31]	Auto exposure value of Camera_0. A write to this field in 'Auto' mode is ignored.
Base + 20h	SUB_GAIN_1	Same format as SUB_GAIN_0		
Base + 24h	SUB_SHUTTER_1	Same format as SUB_SHUTTER_0		
Base + 28h	SUB_AUTO_EXPOSURE_1	Same format as SUB_AUTO_EXPOSURE_0		
Base + 40h	SUB_GAIN_2	Same format as SUB_GAIN_0		

Offset	Name	Field	Bit	Description
Base + 44h	SUB_SHUTTER_2	Same format as SUB_SHUTTER_0		
Base + 48h	SUB_AUTO_EXPOSURE_2	Same format as SUB_AUTO_EXPOSURE_0		
Base + 60h	SUB_GAIN_3	Same format as SUB_GAIN_0		
Base + 64h	SUB_SHUTTER_3	Same format as SUB_SHUTTER_0		
Base + 68h	SUB_AUTO_EXPOSURE_3	Same format as SUB_AUTO_EXPOSURE_0		
Base + 80h	SUB_GAIN_4	Same format as SUB_GAIN_0		
Base + 84h	SUB_SHUTTER_4	Same format as SUB_SHUTTER_0		
Base + 88h	SUB_AUTO_EXPOSURE_4	Same format as SUB_AUTO_EXPOSURE_0		
Base + A0h	SUB_GAIN_5	Same format as SUB_GAIN_0		
Base + A4h	SUB_SHUTTER_5	Same format as SUB_SHUTTER_0		
Base + A8h	SUB_AUTO_EXPOSURE_5	Same format as SUB_AUTO_EXPOSURE_0		

B.12 JPEG_BUFFER_USAGE: 1E84h

This register is for Ladybug cameras only.

Specifies the percentage of the image buffer on the PC that is used for JPEG compressed image data, when the camera is operating in a JPEG mode.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-24]	Reserved
Value	[25-31]	Value. Valid range: 0x00 (0%) to 0x7F (100%) A value of 0 is treated as 0x66 (80%). On Ladybug3 firmware v1.2.2.1 or later, a value of 0 is treated as 0x72 (90%).

B.13 JPEG_CTRL: 1E80h

Specifies the JPEG compression rate.

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
RTP_ON_OFF	[1]	Enable/disable RTP transmission 0: RTP is OFF, 1: RTP is ON
	[2-5]	Reserved

Field	Bit	Description
ON_OFF	[6]	JPEG compression ON_OFF. Read: Read the current status Write: Set the status 0: JPEG compression is OFF, 1: JPEG compression is ON If this bit = 0, other fields will be read only
A_M_Mode	[7]	Read: Read a current mode Write: Set the mode 0: Manual, 1: Automatic JPEG quality control
	[8-23]	Reserved
Value	[24-31]	JPEG quality value. Valid range: 0x01 (1%) to 0x64 (100%). A value of 0 is treated as 60%. A write to this value in ‘Auto’ mode will be ignored.

B.14 JPEG_MAX_QUALITY: 1E8Ch

This register is for Ladybug cameras only.

Specifies the maximum percentage of the JPEG compression when auto compression (JPEG_CTRL: 1E80h) is enabled.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-24]	Reserved
Value	[25-31]	Value. Maximum percentage of JPEG compression when auto compression is enabled. The default value is 80%. A value higher than 80% is not recommended.

B.15 Memory Channel Registers

User Set 0 (or Memory channel 0) stores the factory default settings that can always be restored. Two additional user sets are provided for custom default settings. The camera initializes itself at power-up, or when explicitly reinitialized, using the contents of the last saved user set. Attempting to save user settings to the (read-only) factory default user set causes the camera to switch back to using the factory defaults during initialization.

The values of the following registers are saved in memory channels.

Register Name	Offset
CURRENT_FRAME_RATE	600h
CURRENT_VIDEO_MODE	604h
CURRENT_VIDEO_FORMAT	608h
CAMERA_POWER	610h
CUR_SAVE_CH	620h
BRIGHTNESS	800h

Register Name	Offset
AUTO_EXPOSURE	804h
SHARPNESS	808h
WHITE_BALANCE	80Ch
HUE	810h
SATURATION	814h
GAMMA	818h
SHUTTER	81Ch
GAIN	820h
IRIS	824h
FOCUS	828h
TRIGGER_MODE	830h
TRIGGER_DELAY	834h
FRAME_RATE	83Ch
PAN	884h
TILT	888h
ABS_VAL_AUTO_EXPOSURE	908h
ABS_VAL_SHUTTER	918h
ABS_VAL_GAIN	928h
ABS_VAL_BRIGHTNESS	938h
ABS_VAL_GAMMA	948h
ABS_VAL_TRIGGER_DELAY	958h
ABS_VAL_FRAME_RATE	968h
IMAGE_DATA_FORMAT	1048h
AUTO_EXPOSURE_RANGE	1088h
AUTO_SHUTTER_RANGE	1098h
AUTO_GAIN_RANGE	10A0h
GPIO_XTRA	1104h
SHUTTER_DELAY	1108h
GPIO_STRPAT_CTRL	110Ch
GPIO_CTRL_PIN_x	1110h, 1120h, 1130h, 1140h
GPIO_XTRA_PIN_x	1114h, 1124h, 1134h, 1144h
GPIO_STRPAT_MASK_PIN_x	1118h, 1128h, 1138h, 1148h
MIRROR_IMAGE_CTRL	1054h
FRAME_INFO	12F8h
IMAGE_POSITION	008h
IMAGE_SIZE	00Ch
COLOR_CODING_ID	010h

Register Name	Offset
UDP_PORT	1F1Ch
DESTINATION_IP	1F34h

B.16 ONE_SHOT/MULTI_SHOT: 61Ch

This register allows the user to control single and multi-shot functionality of the camera. During ISO_EN = 1, *One_Shot* = 1 or *Multi_Shot* =1, the register value which reflects the Isochronous packet format cannot change. Data transfer control priority is ISO_EN > One_Shot > Multi_Shot.

Single (One_Shot) transmission is used to transmit the last image without deleting it. Multi-shot transmission is used to transmit the *n* (Count_Number) oldest images and then deletes them.

Format:

Field	Bit	Description
One_Shot	[0]	1 = only one frame of video data is transmitted. (Self cleared after transmission) Ignored if ISO_EN = 1
Multi_Shot	[1]	1 = N frames of video data is transmitted. (Self cleared after transmission) Ignored if ISO_EN = 1 or One_Shot =1
	[2-15]	Reserved.
Count_Number	[16-31]	Count number for Multi-shot function.

B.17 PIXEL_DEFECT_CTRL: 1A60h

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
	[1-5]	Reserved
ON_OFF	[6]	Enable or disable FPGA pixel correction 0: Off, 1: On
	[7]	Reserved
Max_Pixels	[8-19]	Maximum number of pixels that can be corrected by the FPGA
Cur_Pixels	[20-31]	Current number of pixels that are being corrected by the FPGA

B.18 SHUTTER: 81Ch

This register has three states:

State	Description
Manual/Abs	The shutter value is set by the user via the ABS_VAL_SHUTTER register . The *Value* field becomes read only and reflects the converted value of the ABS_VAL_SHUTTER register.
Manual	The user sets the shutter value via the *Value* field. The ABS_VAL_SHUTTER register becomes read only and contains the current shutter time.
Auto	The shutter value is set by the auto exposure controller (if enabled) . Both the *Value* field and the ABS__ VAL_SHUTTER register become read only.

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Abs_Control	[1]	Absolute value control 0: Control with the *Value* field, 1: Control with the Absolute value CSR. If this bit = 1, the value in the *Value* field is ignored.
	[2-4]	Reserved
One_Push	[5]	One push auto mode (controlled automatically by camera only once) Read: 0: Not in operation, 1: In operation Write: 1: Begin to work (self-cleared after operation) If A_M_Mode = 1, this bit is ignored
ON_OFF	[6]	Read: read a status Write: ON or OFF for this feature 0: OFF, 1: ON If this bit = 0, other fields will be read only
A_M_Mode	[7]	Read: read a current mode Write: set the mode 0: Manual, 1: Automatic
High_Value	[8-19]	Upper 4 bits of the shutter value available only in extended shutter mode (outside of specification).
Value	[20-31]	Value. A write to this value in 'Auto' mode will be ignored.

B.19 SOFTWARE_TRIGGER: 62Ch

Note: Bit 0 of this register indicates if the camera is ready to be triggered again for both software and hardware triggering.

Format:

Field	Bit	Description
Software_Trigger	[0]	This bit automatically resets to zero in all trigger modes except Trigger Mode 3. Read: 0: Ready, 1: Busy Write: 0: Reset software trigger, 1: Set software trigger

B.20 TRIGGER_DELAY: 834h

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Abs_Control	[1]	Absolute value control 0: Control with the Value field, 1: Control with the Absolute value CSR. If this bit = 1, the value in the Value field is read-only.
	[2-5]	Reserved
ON_OFF	[6]	Read: read a status Write: ON or OFF for this feature 0: OFF, 1: ON If this bit = 0, other fields will be read only
	[7-19]	Reserved
Value	[20-31]	Value.

B.21 TRIGGER_MODE: 830h

Control of the register is via the *ON_OFF* bit and the *Trigger_Mode* and *Parameter* fields.

Format

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Abs_Control	[1]	Absolute value control 0: Control with the Value field, 1: Control with the Absolute value CSR. If this bit = 1, the value in the Value field is read-only.
	[2-5]	Reserved
ON_OFF	[6]	Read: read a status Write: ON or OFF for this feature 0: OFF, 1: ON If this bit = 0, other fields will be read only
Trigger_Polarity	[7]	Select trigger polarity (except for Software_Trigger) 0: Trigger active low, 1: Trigger active high

Field	Bit	Description
Trigger_Source	[8-10]	Select trigger source: used to select which GPIO pin will be used for external trigger purposes. Sets trigger source ID from *Trigger_Source_Inq* field of TRIGGER_INQ register.
Trigger_Value	[11]	Trigger input raw signal value: used to determine the current raw signal value on the pin. Read only 0: Low, 1: High
	[8-11]	Reserved
Trigger_Mode	[12-15]	Trigger mode (Trigger_Mode_0..15): used to set the trigger mode to be used. For more information, see Asynchronous Trigger Settings. Query the *Trigger_Mode_Inq* fields of the TRIGGER_INQ register for available trigger modes.
	[16-19]	Reserved
Parameter	[20-31]	Parameter for trigger function, if required (optional)

B.22 WHITE_BALANCE: 80Ch

Format:

Field	Bit	Description
Presence_Inq	[0]	Presence of this feature 0: Not Available, 1: Available
Abs_Control	[1]	Absolute value control 0: Control with the Value field, 1: Control with the Absolute Value CSR If this bit is 1, then Value is ignored
	[2-4]	Reserved
One_Push	[5]	One push auto mode (controlled automatically by camera only once) Read: 0: Not in operation, 1: In operation Write: 1: Begin to work (self-cleared after operation) If A_M_Mode = 1, this bit is ignored
ON_OFF	[6]	Read: read a status Write: ON or OFF for this feature 0: OFF, 1: ON If this bit = 0, other fields will be read only
A_M_Mode	[7]	Read: read the current mode. Write: Set the mode. 0: Manual, 1: Auto
U_Value/B_Value	[8-19]	Blue Value. A write to this value in ‘Auto’ mode will be ignored.
V_Value/R_Value	[20-31]	Red Value. A write to this value in ‘Auto’ mode will be ignored.

C Downloads and Support

Teledyne endeavors to provide the highest level of technical support possible to you. Most support resources can be accessed through your product's Support page. From the spherical imaging page, click on your product family and then click the **Go to Support Page** link.

C.1 Finding information

Ladybug® SDK—The Ladybug SDK provides API examples and the LadybugCapPro camera evaluation application. Available from the Ladybug SDK page.

Product Documentation—The camera's *Getting Started Manual* provides information on installing components and software needed to run the camera. The *Technical Reference* provides information on the camera's specifications, features and operations, as well as imaging and acquisition controls. They are available from the downloads tab of the product's support page.

C.2 Contacting technical support

Before contacting Technical Support, have you:

1. Read the product documentation?
2. Searched the knowledge base?
3. Downloaded and installed the latest version of software and/or firmware?

If you have done all the above and still can't find an answer to your question, contact our Technical Support team.

D Legal

View the following from our website:

- Hardware Warranty Information
- Software End User License Agreement
- Third-party open source licenses